Backpack Bluetooth Sync

Developer's Document

Last updated Wednesday, March 12, 2014 by Jennifer Chen

Introduction

Backpack is an Android app which gets the latest news and trending videos on the internet. The content can be viewed within the app and can be spread to other Backpack users without the internet. Our initial effort for content exchange was through bluetooth. This document describes the protocol and API for the bluetooth content synchronization for Backpack Android.

Contents and Metadata

The contents and metadata are being kept in the sdcard/BackpackContent directory. We use protocol buffer for our metadata. See https://developers.google.com/protocol-buffers/ for details. There are currently two files in the content directory; video.dat for keeping the video entries and article.dat for the web entries.

More on protocol buffer

We've created three protocol buffer description files; article.proto, video.proto and comment.proto. These files describe the structure of our metadata. To generate the Java classes from these files, you should have protocol buffer installed and use this command:

```
`protoc -I=src --java_out=src src/article.proto, video.proto, comment.proto`
```

Once the Java classes are generated, don't try to modify these classes. Whenever metadata definition is modified, regenerate these classes with the above command.

Implementation Decisions

Connection

There are two Service classes used to handle the exchange of content data; ConnectionService and ListenerService. The device which initiates the connection creates the ConnectionService and tries to connect to another device. ListenerService is always running and listening for incoming connection, and is on a separate process.

Smart Synching

Currently we are using the most simple way to select which content to be sent to the other device; if the other device does not have the content that I have, send it. In the future we plan

to develop smart synching which uses a scoring system based on user preferences and other factors when determining which entries to send.

Transfer directory

A temporary transfer folder is used for bluetooth sync. It is located inside the content directory; sdcard/BackpackContent/xfer/<RemoteDeviceName>. Each device connected to current device has its own directory under xfer folder. Files under the directory are removed after being merged into the Backpack content.

Buffer size

When sending files over bluetooth, a file should not be sent in one chunk. They must be split into smaller sets.

Reconnect for stability

While sending large files, we ran into stability issues. Bluetooth complains when the receiver keeps receiving data without sending anything. It is also a problem when sender is sending faster than receiver is receiving. In both cases, the transfer hangs forever. We could end the whole synchronization gracefully when problem occurs, but this happens so often we could hardly get any sync to finish completely. We are working on resolving the root of these problems, but as for now we are able to get around the problems by trying to reconnect. Although calling connect on an already connected device throws an exception, it actually re-establishes the flow of the file transfers. We don't loss any of the files which was half-way done when the hanging occurred.

Individual metadata for disconnection

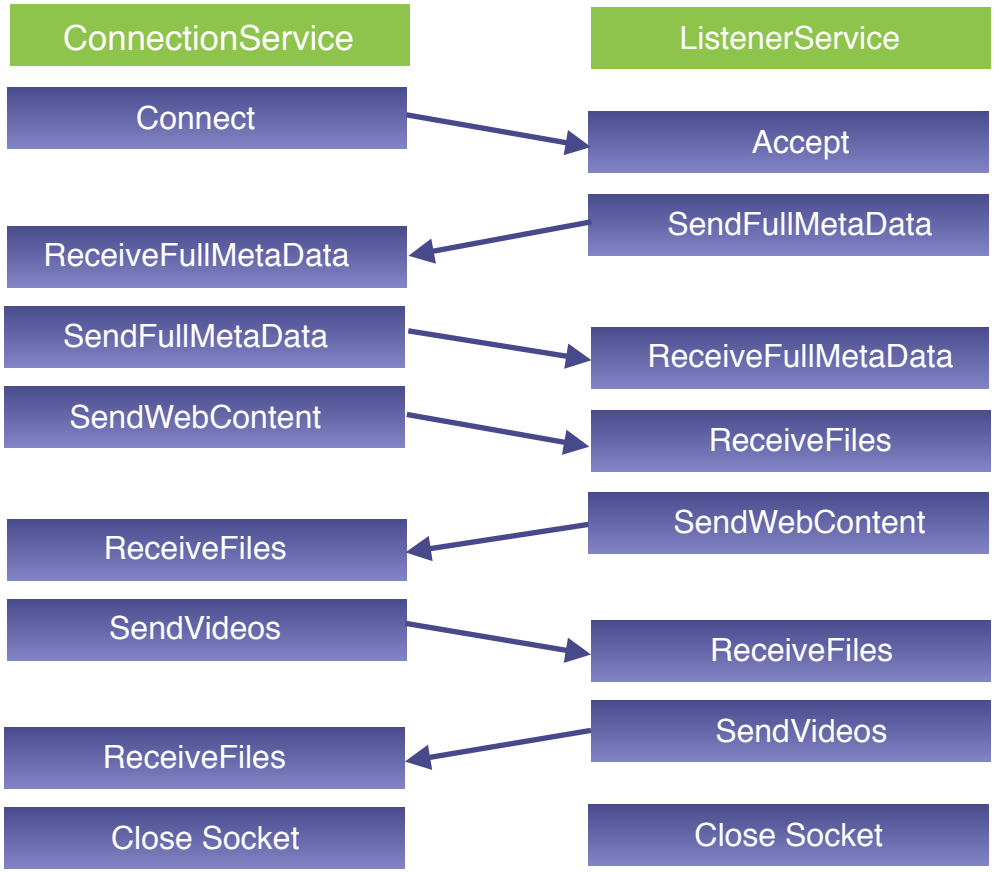
Disconnection happens. When two devices become too far or when a device's bluetooth was suddenly turned off, devices are disconnected. If we wait until for all the contents are transferred and merge everything at once, we could end up with lots of files in the transfer directory but never merged into Backpack content when disconnection occurs. To get around that, we decided to send individual metadata file for each entry. After a content (html or video) file is transferred, a corresponding metadata file is also sent. This way we do not leave lots of content files hanging when disconnection occurs.

Merging transferred file with current content

We have a file monitoring service that monitors the transfer directory for new metadata files. As soon as it sees a new metadata file, it merges that entry with the current content. The user immediately sees the new content on his/her app as soon as it is merged, while the rest of the entries are still being sent. Once a content entry is merged, its related files are removed from the transfer folder.

Before a file transfer is complete, the file name ends with .tmp, which the file observer ignores. Once file transfer is complete, the .tmp is removed so that file observer will handle the merging of the file.

Synchronization Protocol



API Details

edu.isi.backpack.services.ListenerService

ListenerService is a Service class that runs on its own process to listen to incoming sync requests. It is started as soon as the phone starts and is kept alive until the phone is turned off.

edu.isi.backpack.services.ConnectionService

ConnectionService is a Service class and is created when a Backpack user wants to sync the contents with another device. This service is destroyed once the sync is complete.

edu.isi.backpack.bluetooth.InfoMessage

This class defines a message packet used to exchange information between devices.

edu.isi.backpack.bluetooth.MessageHandler

This class defines methods for handling messages and data between devices

SendFullMetaData(short, File)

- Sends an InfoMessage about the metadata file about to be sent.
- Sends the actual file.
- Waits for an ACK InfoMessage from the other device.

ReceiveFullMetaData(File)

- Waits for an InfoMessage about the metadata about to be received.
- Receives the actual file.
- Compare the metadata with its own metadata and calculate which entries the other device needs. See Smart Synching.
- Sends ACK InfoMessage indicating file received.

SendWebContent(File)

- Sends an InfoMessage indicating the start of a bulk transfer operation.
- Waits for an ACK InfoMessage from the other device.
- For each web entry to be sent
  - Sends InfoMessage about the file
  - Sends the html file
  - Waits ACK from receiver
  - For each image belonging to the html

        - Sends InfoMessage about the image
        - Sends image file
        - Waits for ACK from receiver
    - Sends the metadata for this individual entry
        - Sends InfoMessage about the file
        - Sends file
        - Waits for ACK from receiver
- Sends InfoMessage indicating end of bulk transfer.
- Waits for ACK from receiver.

SendVideos(File)

- Sends an InfoMessage indicating the start of a bulk transfer operation.
- Waits for an ACK InfoMessage from the other device.
- For each video entry to be sent,
    - Sends InfoMessage about the file
    - Sends the video file
    - Waits for an ACK from receiver
    - Sends bitmap thumbnail of the video
        - Sends InfoMessage about the file
        - Sends bitmap image
        - Waits for ACK from receiver
    - Sends the metadata for this individual entry
        - Sends InfoMessage about the file
        - Sends file
        - Waits for ACK from receiver
- Sends InfoMessage indicating end of bulk transfer.
- Waits for ACK from receiver.

ReceiveFiles(File)

- Waits for InfoMessage indicating bulk transfer.
- Sends ACK InfoMessage to sender.
- Reads incoming data and create local files.

edu.isi.backpack.bluetooth.Connector

This class defines methods for implementing data transaction used by MessageHandler.

edu.isi.backpack.services.FileMonitorService

This service class maintains the CustomFileObserver that monitors the transfer directory.

edu.isi.backpack.CustomFileObserver

This is a FileObserver class which monitors the transfer directory sdcard/BackpackContent/xfer/. It ignores any .tmp files, which are files still being transferred. When a new metadata file is created, it calls ContentManagementTask to merge the new content.

edu.isi.backpack.tasks.ContentManagementTask

This class has static methods MergeArticles and MergeVideos to merge one content directory into another. It compares the source metadata with the destination metadata files and adds any non-existing entries into destination.