

Updating files via tinyssb

```

master feed
+-----+-----+-----+...
| Head | mk_child | mk_child |
+-----+-----+-----+...
                :           |   update feed
                :           | +-----+-----+-----+...
node feed      '--> | Head | is_child | mk_child| mk_child |
                   +-----+-----+-----+...
                                     |
/------'
|
|   version control feed
|   +-----+-----+-----+...
|--> | Head | is_child | apply_up|
    +-----+-----+-----+...
                                     |
/------
|
|   file_1 update feed
|   +-----+-----+-----+...
|--> | Head | update | mk_child | update blob 1 |
    +-----+-----+-----+...
                                   |   file_1 emergency feed
                                   |   +-----+-----+
                                   |--> | Head | is_child |
                                       +-----+

```

```
Packet type updfile: payload
```

```

      <- - - - - 47B - - - - ->
      | var_int len | file_name | version number | padding |
+-----+-----+-----+-----+
| 1B          | 4B          |              |            |
+-----+-----+-----+-----+
Packet type applyup: payload
      32B          4B          12B
<- - - - -> <- - - - -> <- - - - ->
+-----+-----+-----+-----+
| file update feed fid | update sequence number | padding |
+-----+-----+-----+-----+

```

This allows for a file name of maximum 47B.

Version control feeds can easily be managed, by maintaining a Python dictionary:

```
vc_dict = {  
    file_name_1: (feed_1, emergency_feed_1),  
    file_name_2: (file_2, emergency_feed_2),  
    ...  
}
```

Once an emergency feed is triggered, the information can easily be updated:

```
new_emergency_feed = feed_manager.create_child_feed(old_emergency_feed)  
new_emergency_feed.append_version_control(file_name)  
vc_dict[file_name] = (old_emergency_feed, new_emergency_feed)
```