Publish/playback channel quality control

Since build 2.6.110it is possible to receive messages containing current publishing or playback bitrate from server and to control channel quality based on client and server bitrates difference.

To enable channel quality control, use the method `WCSStream.enableConnectionQualityCalculation()`

```
publishStream?.enableConnectionQualityCalculation(true);
```

Then register a callback function to get quality metric and current client and server bitrate values averaged by Kalman filter

```
publishStream?.onConnectionQualityCallback({currentQuality, clientFiltered, serverFiltered in
    self.updateQualityStatus(currentQuality, view: self.publishQuality);
});
```

Usage example:

codeof handler registration, codeof handler

```
@IBOutlet weak var publishQuality: UILabel!
...
publishStream?.enableConnectionQualityCalculation(true);
publishStream?.onConnectionQualityCallback({currentQuality, clientFiltered, serverFiltered in
    self.updateQualityStatus(currentQuality, view: self.publishQuality);
});
...
fileprivate func updateQualityStatus(_ quality:kFPWCSConnectionQuality, view: UILabel) {
    switch (quality) {
    case .fpwcsConnectionQualityBad:
        view.text = "BAD";
        view.textColor = .red;
        break;
    case .fpwcsConnectionQualityGood:
        view.text = "GOOD";
        view.textColor = .yellow;
        break;
    case .fpwcsConnectionQualityPerfect:
        view.text = "PEFRECT";
        view.textColor = .green;
        break;
    case .fpwcsConnectionQualityUnknown:
        view.text = "UNKNOWN";
        view.textColor = .darkText;
    }
}
```