## A Operational Profile Data

In this section we present operational profile data for one of the FL populations that are currently active in the deployed FL system, augmenting the discussion in Sec. 9. The subject FL population primarily comes from the same time zone.

Fig. 6 illustrates how availability of the devices varies through the day and its impact on the round completion rate. Because the FL server schedules an FL task for execution only once deserired number of devices are available and selected the round completion rate oscillates in sync with the devices availability.

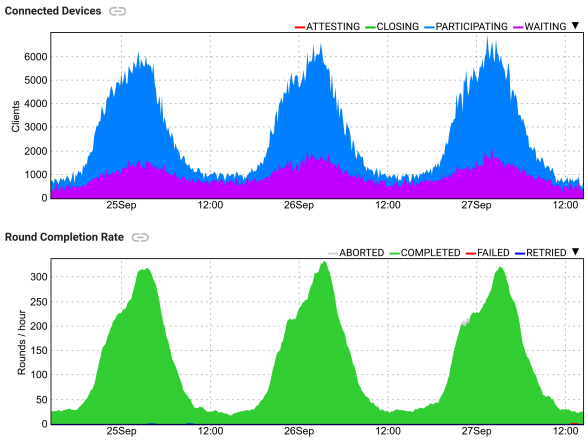


*Figure 6.* A subset of the connected devices over three days (top) in states "participating" (blue) and "waiting" (purple). Other states ("closing" and "attesting") are too rare to be visible in this graph. The rate of successful round completions (green, bottom) is also shown, along with the rate of other outcomes ("failure", "retry", and "abort") plotted on the same graph but too low to be visible.

Fig. 7 illustrates the average number of devices participating in an FL task round and the outcomes of the participation. Note that in each round the FL server selects more devices for the participation than desired to complete to offset the devices that drop out during execution. Therefore in each round there are devices that were aborted after a desired number of devices successfully complete. Another noteworthy aspect is drop out rate correlation with the time of day, specifically the drop out rate is higher during the day time compared to the night time. This is explained by higher probability of the device eligibility critiria changes due interaction with a device.

Fig. 8 shows distribution of round run and device participation time. There are two noteworthy overvations. First is that the round run time is roughly equal to the majority of the device participation time which is explained by the fact that the FL server selects more than needed devices for participation and stops execution when enough devices complete. Second is that device participation time is capped.

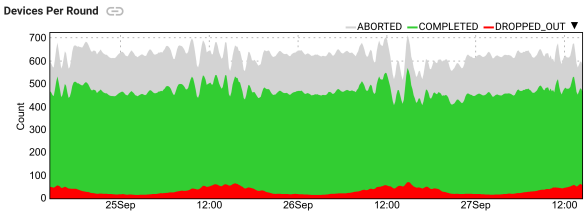


*Figure 7.* Average number of devices completed, aborted and dropped out from round execution

This is a mechanism used by the FL server to deal with the straggler devices. In other words the round run time capped by the server.

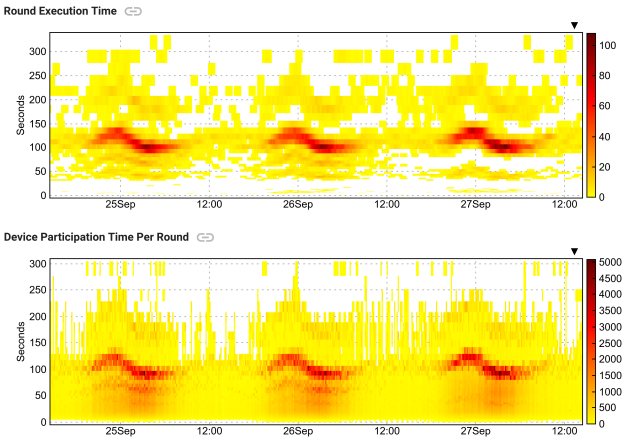


*Figure 8.* Round exection and device participation time

Fig. 9 illustrates assymetry in server network traffic, specifically that download from server dominates upload. There are several aspects that contribute. Namely each device downloads both an FL task plan and current global model (plan size is comparable with the global model) whereas it uploads only updates to the global model; the model updates inherently more compressible compared to the global model.

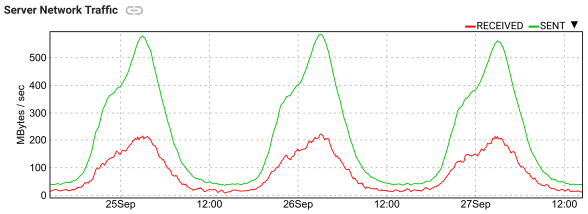


*Figure 9.* Server network traffic

Tab. 1 shows the training round session shape visualizations, generated from the clients' training state event logs. As shown, 75% of clients complete their training rounds suc-

| Session Shape | Count | Percent |
|---|---|---|
| -v[]+^ | 1,116,401 | 75% |
| -v[]+# | 327,478 | 22% |
| -v[! | 29,771 | 2% |

*Table 1.* Distribution of on-device training round sessions. Legend: - = FL server checkin, v = downloaded plan, [ = training started, ] = training completed, + = upload started, ^ = upload completed, # = upload rejected, ! = interrupted.

cessfully, 22% of clients complete their training rounds, but have their results rejected by the server (these are the devices which report back after the reporting window already closed), and 2% of clients are interrupted before being able to complete their round (e.g. because the device exited the idle state).

## B FEDERATED AVERAGING

In this section, we show the Federated Averaging algorithm from McMahan et al. (2017) for the interested reader.

**Algorithm 1** `FederatedAveraging` targeting updates from $K$ clients per round.

**Server executes:**
initialize $w_0$
**for** each round $t = 1, 2, \dots$ **do**
  Select $1.3K$ eligible clients to compute updates
  Wait for updates from $K$ clients (indexed $1, \dots, K$)
  $(\Delta^k, n^k) = \text{ClientUpdate}(w)$ from client $k \in [K]$.
  $\bar{w}_t = \sum_k \Delta^k$ *// Sum of weighted updates*
  $\bar{n}_t = \sum_k n^k$ *// Sum of weights*
  $\Delta_t = \Delta_t^k / \bar{n}_t$ *// Average update*
  $w_{t+1} \leftarrow w_t + \Delta_t$

**ClientUpdate**$(w)$**:**
$\mathcal{B} \leftarrow$ (local data divided into minibatches)
$n \leftarrow |\mathcal{B}|$ *// Update weight*
$w_{\text{init}} \leftarrow w$
**for** batch $b \in \mathcal{B}$ **do**
  $w \leftarrow w - \eta \nabla \ell(w; b)$
$\Delta \leftarrow n \cdot (w - w_{\text{init}})$ *// Weighted update*
*// Note $\Delta$ is more amenable to compression than $w$*
return $(\Delta, n)$ to server