Federated Learning in Mobile Edge Networks: A Comprehensive Survey

Filipe Maciel - Seminários de tópicos em sistemas distribuídos (federated learning)

Objetivos do artigo

---

1. Federated Learning sobre Multi-Access Edge Computing;
    a. Ficará de fora da apresentação a parte sobre segurança e privacidade.
2. Federated Learning para Multi-Access Edge Computing.

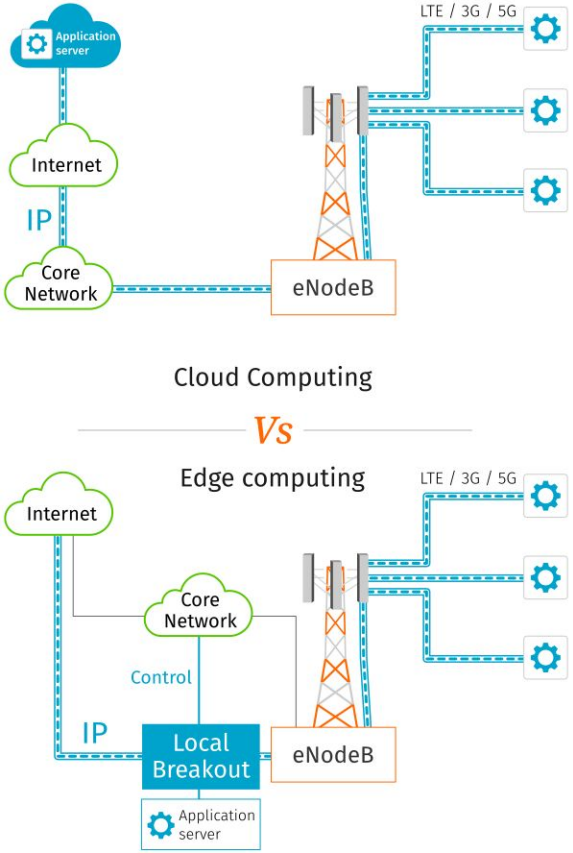
Agenda

---

1. O que é Multi-Access Edge Computing?
2. Custo de comunicação do FL;
3. Alocação de recursos para FL.

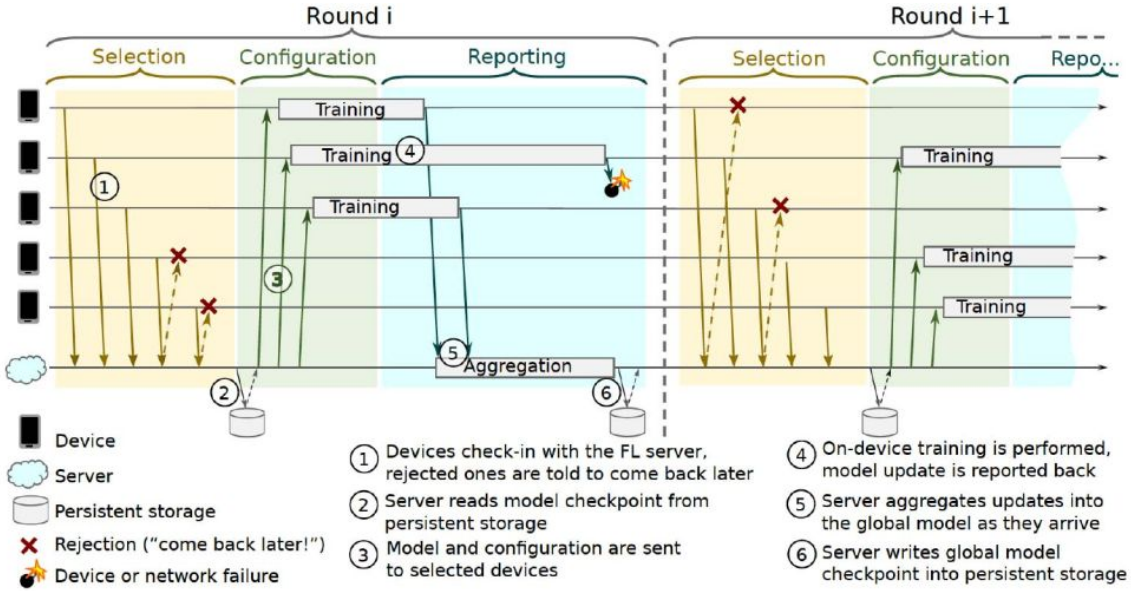
O que é multi-access edge computing?

- Infra estrutura que trás o tráfego e os serviços próximos ao consumidor/gerador.
  - Processamento local.
    - Reações mais rápidas às requisições dos usuários;
    - Redução de custos para apps de baixa latência;
    - Evita tráfego no núcleo da rede.



Custos de comunicação do FL

Protocolo de treinamento

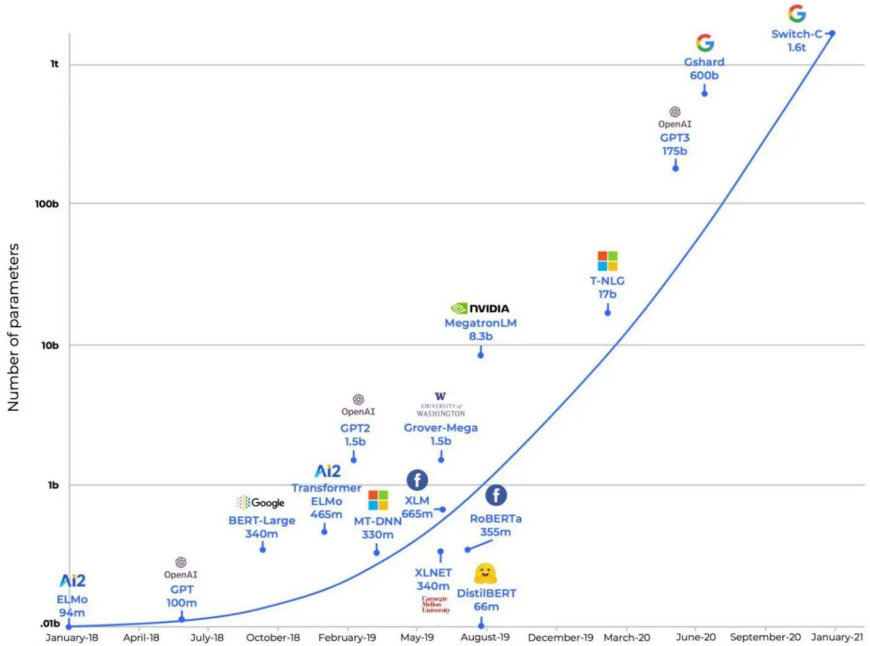


Fonte: [Towards Federated Learning at Scale: System Design (MLSys 2019)]

Custos de comunicação do FL

O protocolo não é adequado para:

1. Cenários assíncronos;
2. Agregação baseada em otimização diferente de SGD-Batch;
3. Outras topologias que não estrela;
4. Enlaces assimétricos;
5. Meios de comunicação não confiáveis.



Custos de comunicação do FL

Técnicas de redução de custo:

- Computação na borda ou nos dispositivos finais:
  - Aumentar essa computação para diminuir a quantidade de rodadas de treino.
  - Desenvolver algoritmos de convergência mais rápida.
- Compressão de modelo:
  - Transformar o modelo em uma versão mais compacta, sem reduzir a qualidade do treino.

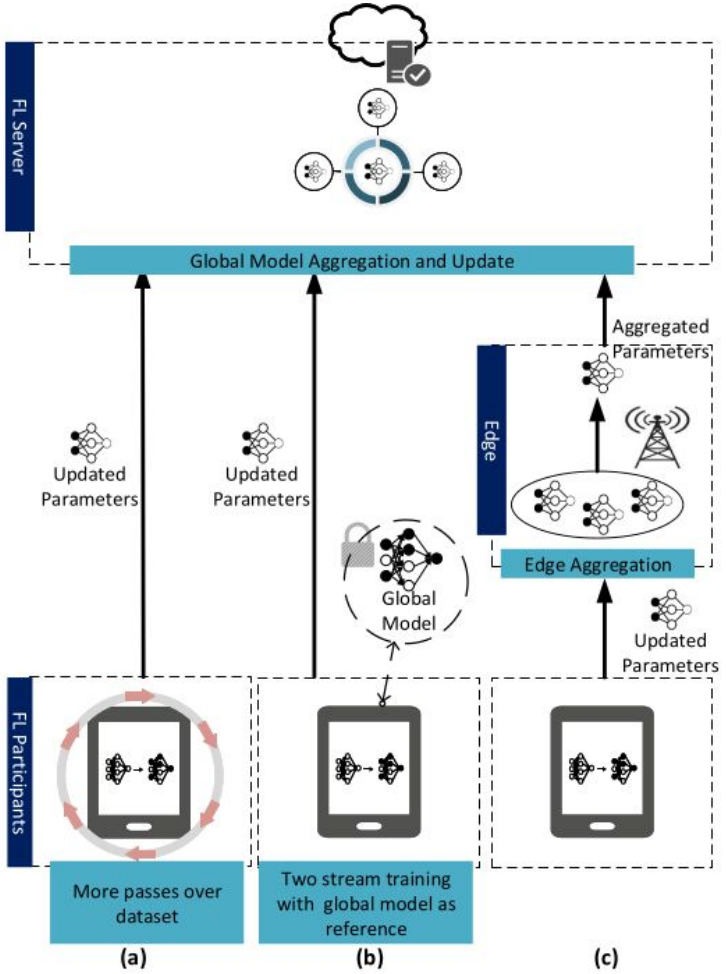
Técnicas de redução de custo:

- Atualização baseada em importância:
  - Selecionar os modelos locais de maior relevância, ou apenas pesos, para serem transmitidos na rodada.

Custos de comunicação do FL

Computação na borda ou nos dispositivos finais:

A. Aumento da computação nos dispositivos;
B. Treinamento em dois fluxos com modelo global como referência;
C. Agregação em servidor edge intermediário.



Custos de comunicação do FL

FedAvg

HierFAVG

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

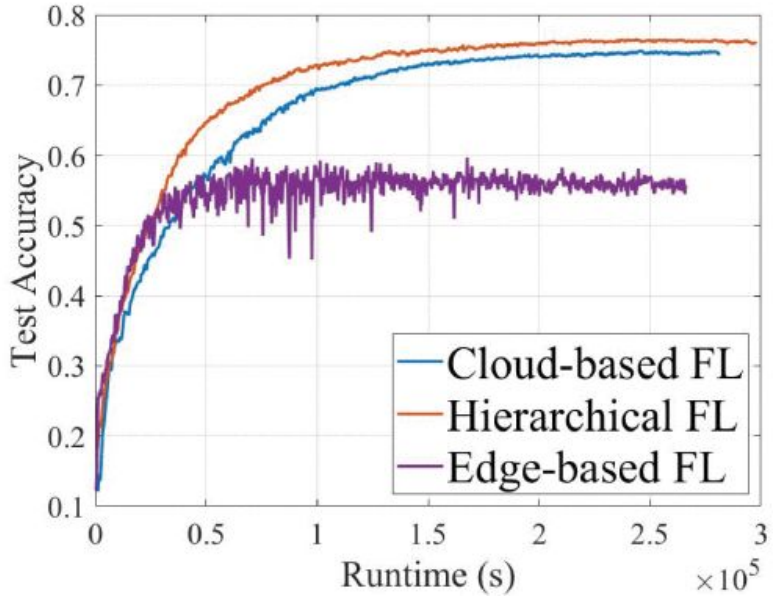
ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Algorithm 1: Hierarchical Federated Averaging (HierFAVG)

procedure HIERARCHICALFEDERATEDAVERAGING
  Initialized all clients with parameter $\boldsymbol{w}_0$
  for $k = 1, 2, \dots K$ **do**
    for each client $i = 1, 2, \dots, N$ in parallel **do**
      $\boldsymbol{w}_i^\ell(k) \leftarrow \boldsymbol{w}_i^\ell(k-1) - \eta \nabla F_i(\boldsymbol{w}_i^\ell(k-1))$
    end for
    if $k \mid \kappa_1 = 0$ **then**
      for each edge $\ell = 1, \dots, L$ in parallel **do**
        $\boldsymbol{w}^\ell(k) \leftarrow \text{EdgeAggregation}(\{\boldsymbol{w}_i^\ell(k)\}_{i \in \mathcal{C}^\ell})$
        if $k \mid \kappa_1 \kappa_2 \neq 0$ **then**
          for each client $i \in \mathcal{C}^\ell$ in parallel **do**
            $\boldsymbol{w}_i^\ell(k) \leftarrow \boldsymbol{w}^\ell(k)$
          end for
        end if
      end for
    end if
    if $k \mid \kappa_1 \kappa_2 = 0$ **then**
      $\boldsymbol{w}(k) \leftarrow \text{CloudAggregation}(\{\boldsymbol{w}^\ell(k)\}_{\ell=1}^L)$
      for each client $i = 1 \dots N$ in parallel **do**
        $\boldsymbol{w}_i^\ell(k) \leftarrow \boldsymbol{w}(k)$
      end for
    end if
  end for
end procedure
function EDGEAGGREGATION$(\ell, \{\boldsymbol{w}_i^\ell(k)\}_{i \in \mathcal{C}^\ell})$ *//Aggregate locally*
  $\boldsymbol{w}^\ell(k) \leftarrow \frac{\sum_{i \in \mathcal{C}^\ell} |\mathcal{D}_i^\ell| \boldsymbol{w}_i^\ell(k)}{|\mathcal{D}^\ell|}$
  return $\boldsymbol{w}^\ell(k)$
end function
function CLOUDAGGREGATION$(\{\boldsymbol{w}^\ell(k)\}_{\ell=1}^L)$ *//Aggregate globally*
  $\boldsymbol{w}(k) \leftarrow \frac{\sum_{\ell=1}^L |\mathcal{D}^\ell| \boldsymbol{w}^\ell(k)}{|\mathcal{D}|}$
  return $\boldsymbol{w}(k)$
end function

Custos de comunicação do FL

Comparação acurácia x tempo de execução:



Custos de comunicação do FL

Compressão de modelos:

- Poda (“esparsificação”): redução de conexões (pesos) na rede.
  - Exclui-se parte das conexões de um neurônio, todo o neurônio e até camadas inteiras.
- Subamostragem: formação de um subconjunto aleatório da rede.
- Quantização: redução de bits para representação dos pesos em uma rede.
  - Redução de pontos flutuantes de 64 ou 32 bits para representação em 16 bits, 8 bits ou até menores.

Custos de comunicação do FL

FEDZIP: A Compression Framework for Communication-Efficient Federated Learning

1. Poda baseada em Top-z;
2. Quantização com k-means;
3. Compressão com codificação de huffman.

Resultados: atinge taxas de compressão de até 1085× e preserva até 99% de largura de banda e 99% de energia para clientes durante a comunicação.

Custos de comunicação do FL

Fedzip:

Algorithm 1 FedZip. The clients are indexed by m; P_m is a set of data for the mth client, E is the number of local epochs, C_M is the number of all clients for FedAvg, η is the learning rate, and B is the local mini-batch size.

procedure SERVER EXECUTION:
  initialize $w_{t=0}$
  for round $t = 1,2,...$ **do**
    $m \leftarrow max(C.M, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for client mth $\in S_t$ *in parallel* **do**
      $msg^m_{t+1}, \theta^m_{t+1} \leftarrow ClientUpdate(m, w_t)$
      $w^m_{t+1} \leftarrow decoding(msg^m_{t+1}, \theta^m_{t+1})$
    end for
  end for
  $w_{t+1} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} w^m_{t+1}$
end procedure
procedure CLIENT UPDATE(M, W):
  $B \leftarrow$ (split P_m into batches of size B)
  for local epoch i from 1 to E **do**
    for batch $b \in B$ **do**
      $w \leftarrow w - \eta \nabla l(w, b)$
    end for
  end for
  encoding(Δw) :
    $msg^m_{t+1}, \theta^m_{t+1} \leftarrow encoding(w)$
  return $msg^m_{t+1}, \theta^m_{t+1}$ to the Server
end procedure

Custos de comunicação do FL

Fedzip:

Algorithm 2 Encoding framework; c_j refers to the jth centroid among clusters. As it is mentioned, $|c_j| = 3$ means we have three centroids.

procedure ENCODING:
  Sparsification
    $w \leftarrow top\text{-}z(w)$
  Quantization
    $w \leftarrow K\text{-}means(w)$
  for each Centroid of $w_m \in w$ **do**
    $w_i \leftarrow c_j$
  end for
  Encoding
  select one of the methods below to build the update message
    1-$msg^m_{t+1}, \theta \leftarrow$ Huffman(w)
    2-$msg^m_{t+1}, \theta \leftarrow$ Exact Position(w)
    3-$msg^m_{t+1}, \theta \leftarrow$ Difference of Address Position(w)[2]
  send the decoding table
end procedure
procedure SERVER:
  Decoding
    $w \leftarrow$ decoding(msg, θ)
end procedure

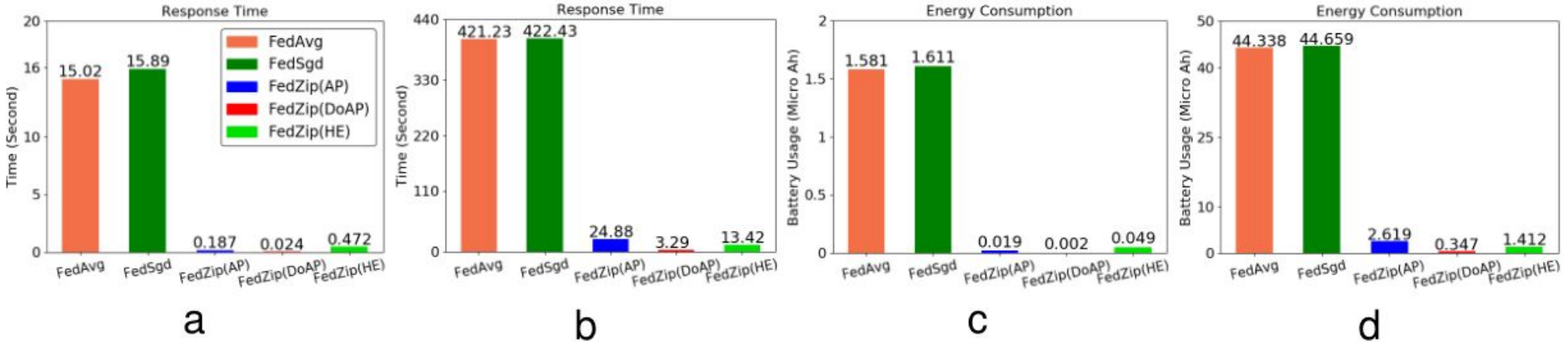
Custos de comunicação do FL

Fedzip:

Methods	Model	Convergence Speed	Training Accuracy	Test Accuracy	Loss	Size of Updates (MB)[1]	Compression Rate	Number of Clients and C	Round (N)
FedAvg	CNN	baseline	99.44	**98.03**	**0.013**	4.79	1x E=1, B=32	50, C=1	20
FedSGD	CNN	low	99.04	97.65	0.031	4.79	1x E=1, B=32	50, C=0	100
FedZip	CNN	same	99.34	97.79	0.026	**0.0078**	Up to **1085**x E=1, B=32	50, C=1	20
FedAvg	VGG16	baseline	99.82	**94.82**	**0.1708**	134.54	1x E=1, B=32	50, C=1	20
FedSGD	VGG16	low	99.02	92.59	0.6213	134.54	1x E=1, B=32	50, C=0	100
FedZip	VGG16	same	99.42	93.30	0.5719	**0.6911**	Up to **194**x E=1, B=32	50, C=1	20

Custos de comunicação do FL

Fedzip:



Custos de comunicação do FL

Atualização baseada em importância:

- Atualização apenas de pesos relevantes:
  - eSGD: Communication Efficient Distributed Deep Learning on the Edge
- Atualização de modelos relevantes:
  - CMFL: Mitigating Communication Overhead for Federated Learning

Custos de comunicação do FL

CMFL:

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \sum_{k=1}^{D} \eta_k \nabla f_k(\mathbf{x}_t) = \mathbf{x}_{t-1} + \sum_{k=1}^{D} \mathbf{u}_{k,t}$$

$$e(\mathbf{u}, \bar{\mathbf{u}}) = \frac{1}{N} \sum_{j=1}^{N} I(\mathrm{sgn}(u_j) = \mathrm{sgn}(\bar{u}_j))$$

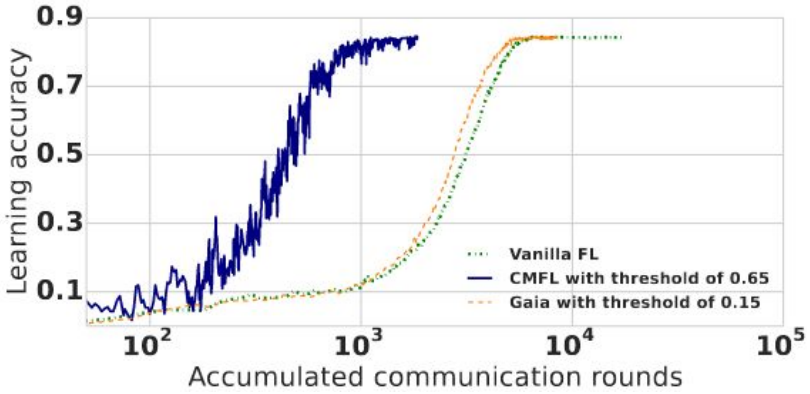
$$I(\mathrm{sgn}(u_j) = \mathrm{sgn}(\bar{u}_j)) = 1$$

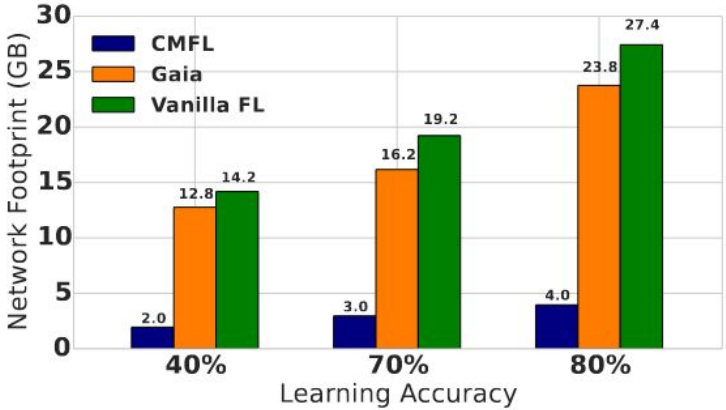
Algorithm 1 Communication-Mitigated FL

procedure GLOBALOPTIMIZATION
  Input: Client set $\mathbb{C} = \langle c_1, \ldots, c_D \rangle$
  Initialize the global model $\mathbf{x}_0$ and the global update $\bar{\mathbf{u}}_0$
  for each iteration $t = 1, 2, \ldots$ **do**
    for all client $c_k \in \mathbb{C}$ **do in parallel**
      $(s_{k,t}, u_{k,t}) \leftarrow$ LocalUpdate $(k, \mathbf{x}_{t-1}, \bar{\mathbf{u}}_{t-1})$
    $\mathbb{S}^t \leftarrow \{\mathbf{u}_{k,t} | \quad s_{k,t}$ is True$\}$ ▷ relevant updates
    $\bar{\mathbf{u}}_t \leftarrow \frac{1}{|\mathbb{S}^t|} \sum_{\mathbf{u}_{k,t} \in \mathbb{S}^t} \mathbf{u}_{k,t}$ ▷ global update
    $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \bar{\mathbf{u}}_t$
procedure LOCALUPDATE
  Input: Client index k, Model $\mathbf{x}_{t-1}$ and Update $\bar{\mathbf{u}}_{t-1}$
  Execute the local training and obtain the local update $\mathbf{u}_{k,t}$
  $s_{k,t} \leftarrow$ CheckRelevance $(\bar{\mathbf{u}}_{t-1}, \mathbf{u}_{k,t})$
  if $s_{k,t}$ is False **then**
    $\mathbf{u}_k \leftarrow$ NULL ▷ exclude irrelevant update
  return $(s_{k,t}, \mathbf{u}_{k,t})$
procedure CHECKRELEVANCE
  Input: Global update $\bar{\mathbf{u}}_{t-1}$ and Client-side update $\mathbf{u}_{k,t}$
  Calculate the relevance $e(\mathbf{u}_{k,t}, \bar{\mathbf{u}}_{t-1})$ following Eq. (9)
  if $e(\mathbf{u}_{k,t}, \bar{\mathbf{u}}_{t-1}) < v(t)$ **then**
    return True
  else
    return False ▷ identify irrelevant updates

Custos de comunicação do FL

CMFL:





Custos de comunicação do FL

Resumo:

- Custos de comunicação devem ser considerados para FL em grande escala, para tecnologias de comunicação com baixa largura de banda e dispositivos com restrições de energia;
- Técnicas:
  - Reduzir a quantidade de bits transmitidos por atualização ou reduzir atualizações de modelos do cliente para servidor;
- Considerar custo benefício dessa redução do custo com a acurácia;

Alocação de recursos para FL

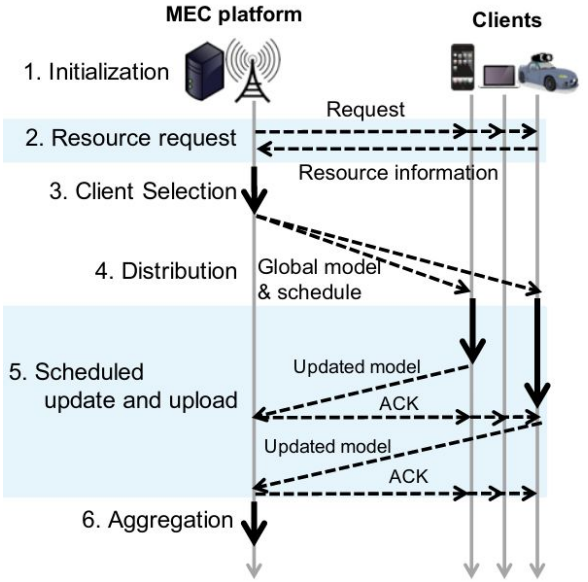
- Seleção de participantes:
  - Treinamento tem gargalo no dispositivo mais lento. Então, objetivo é selecionar subconjunto de participantes que minimize esse gargalo.
- Gestão conjunta de recursos de computação e rádio:
  - Desenvolvimento de novas tecnologias de rádio que favoreçam o FL.
  - Não será comentado.
- Agregação adaptativa:
  - Adaptar a frequência de agregações para incrementar a eficiência do treino em condições de restrição de recursos.
- Mecanismos de incentivo:
  - Dispositivos podem se negar a participar de treinamento porque consome recursos. Mecanismos de incentivo à colaboração devem ser pensados.

Alocação de recursos para FL

[Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge]

Protocol 2 Federated Learning with Client Selection. K is the number of clients, and $C \in (0, 1]$ describes the fraction of random clients that receive a resource request in each round.

1: `Initialization` in Protocol 1.
2: `Resource Request`: The MEC operator asks $\lceil K \times C \rceil$ random clients to participate in the current training task. Clients who receive the request notify the operator of their resource information.
3: `Client Selection`: Using the information, the MEC operator determines which of the clients go to the subsequent steps to complete the steps within a certain deadline.
4: `Distribution`: The server distributes the parameters of the global model to the selected clients.
5: `Scheduled Update and Upload`: The clients update global models and upload the new parameters using the RBs allocated by the MEC operator.
6: `Aggregation` in Protocol 1.
7: All steps but `Initialization` are iterated for multiple rounds until the global model achieves a desired performance or the final deadline arrives.



Alocação de recursos para FL

Seleção: o máximo de clientes possível dentro de um limite de tempo (Distribution+Update+Upload).

$$\Theta_i := \begin{cases} 0 & \text{if } i = 0; \\ T_i^{\mathrm{UD}} + T_i^{\mathrm{UL}} & \text{otherwise,} \end{cases}$$

$$T_i^{\mathrm{UD}} = \sum_{j=1}^{i} \max\{0, t_{k_j}^{\mathrm{UD}} - \Theta_{j-1}\},$$

$$T_i^{\mathrm{UL}} = \sum_{j=1}^{i} t_{k_j}^{\mathrm{UL}}.$$

$$\max_{\mathbb{S}} \quad |\mathbb{S}|$$

$$\text{s.t.} \quad T_{\mathrm{round}} \geq T_{\mathrm{cs}} + T_{\mathbb{S}}^{\mathrm{d}} + \Theta_{|\mathbb{S}|} + T_{\mathrm{agg}}.$$

Algorithm 3 `Client Selection` in Protocol 2

Require: Index set of randomly selected clients $\mathbb{K}'$

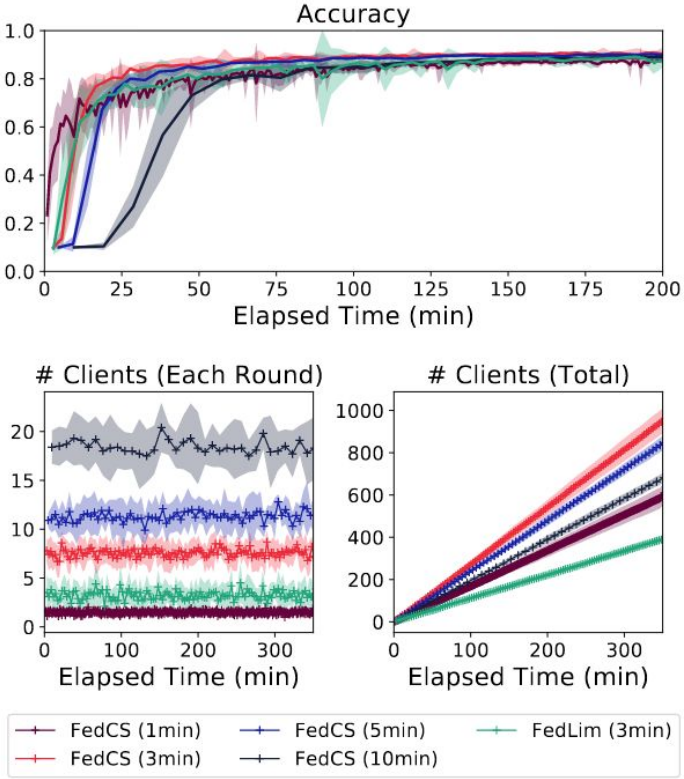
Initialization $\mathbb{S} \leftarrow \{\}$, $T_{\mathbb{S}=\emptyset}^{\mathrm{d}} \leftarrow 0$, $\Theta \leftarrow 0$
while $|\mathbb{K}'| > 0$ **do**
  $x \leftarrow \arg\max_{k \in \mathbb{K}'} \frac{1}{T_{\mathbb{S}\cup k}^{\mathrm{d}} - T_{\mathbb{S}}^{\mathrm{d}} + t_k^{\mathrm{UL}} + \max\{0, t_k^{\mathrm{UD}} - \Theta\}}$
  remove x from $\mathbb{K}'$
  $\Theta' \leftarrow \Theta + t_x^{\mathrm{UL}} + \max\{0, t_x^{\mathrm{UD}} - \Theta\}$
  $t \leftarrow T_{\mathrm{cs}} + T_{\mathbb{S}\cup x}^{\mathrm{d}} + \Theta' + T_{\mathrm{agg}}$
  if $t < T_{\mathrm{round}}$ **then**
    $\Theta \leftarrow \Theta'$
    add x to $\mathbb{S}$
  end if
end while
return $\mathbb{S}$

Alocação de recursos para FL

IID

Method	CIFAR-10		
	ToA@0.5	ToA@0.75	Accuracy
FedLim ($T_{\text{round}} = 3$ min)	38.1	209.2	0.77
FedCS			
$T_{\text{round}} = 3$ min ($r = 0\%$)	**25.8**	**132.7**	**0.79**
$T_{\text{round}} = 3$ min ($r = 10\%$)	**27.9**	**138.1**	**0.78**
$T_{\text{round}} = 3$ min ($r = 20\%$)	**31.1**	**178.3**	**0.78**
$T_{\text{round}} = 1$ min ($r = 0\%$)	NaN	NaN	0.50
$T_{\text{round}} = 5$ min ($r = 0\%$)	41.0	166.6	0.79
$T_{\text{round}} = 10$ min ($r = 0\%$)	75.7	281.7	0.76

Method	Fashion-MNIST		
	ToA@0.5	ToA@0.85	Accuracy
FedLim ($T_{\text{round}} = 3$ min)	10.4	66.8	0.90
FedCS			
$T_{\text{round}} = 3$ min ($r = 0\%$)	**10.6**	**33.5**	**0.91**
$T_{\text{round}} = 3$ min ($r = 10\%$)	**11.3**	**32.1**	**0.92**
$T_{\text{round}} = 3$ min ($r = 20\%$)	**12.7**	**37.0**	**0.91**
$T_{\text{round}} = 1$ min ($r = 0\%$)	3.0	73.7	0.89
$T_{\text{round}} = 5$ min ($r = 0\%$)	18.1	48.8	0.92
$T_{\text{round}} = 10$ min ($r = 0\%$)	42.0	93.3	0.91

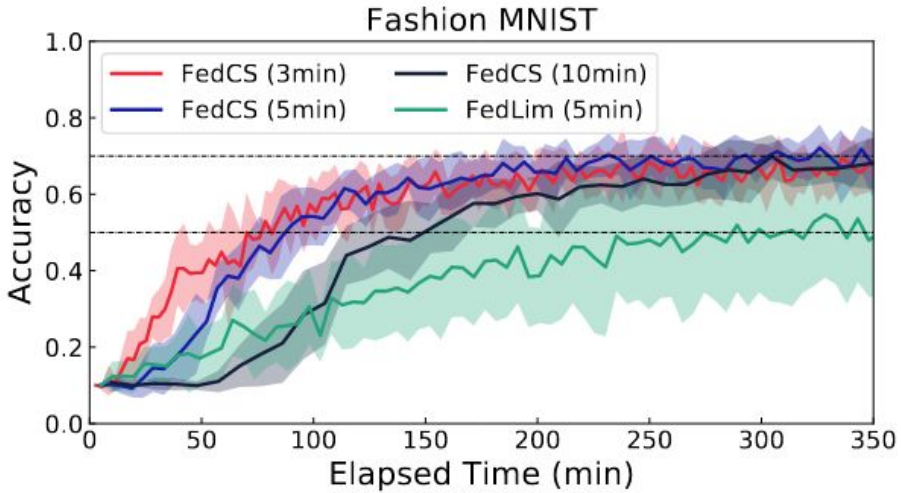


Alocação de recursos para FL

Non-iid

Method	CIFAR-10		
	ToA@0.35	ToA@0.5	Accuracy
FedLim ($T_{\text{round}} = 5$ min)	NaN	NaN	0.31
FedCS ($T_{\text{round}} = 5$ min)	**91.7**	**213.7**	**0.54**

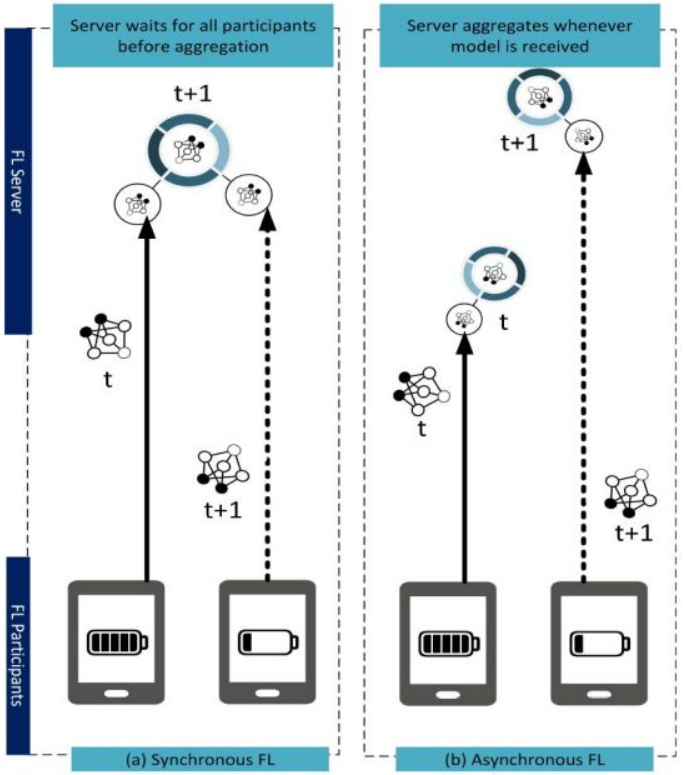
Method	Fashion-MNIST		
	ToA@0.5	ToA@0.7	Accuracy
FedLim ($T_{\text{round}} = 5$ min)	NaN	NaN	0.46
FedCS ($T_{\text{round}} = 5$ min)	**82.4**	**187.7**	**0.71**



Alocação de recursos para FL

Agregação adaptativa:

- FL síncrona: Agregação somente após todos os updates.
- FL assíncrona: Agregação à medida que updates chegam.
  - Ainda tem problemas de convergência.



Alocação de recursos para FL

Adaptive Federated Learning in Resource Constrained Edge Computing Systems:

- Um algoritmo para determinar a frequência de agregação global para que o recurso disponível seja usado com mais eficiência.
  - 3 fases: modelo local, agregação em edge, agregação global.
    - Realizar a agregação global após um certo número de agregações locais.
      - Algoritmo adapta a quantidade de agregações intermediárias para execução dentro de recursos limitados no servidor.
  - Melhora modestamente a acurácia e a perda, dentro do limite efetivo de recursos.
    - Realiza prova da convergência dentro desse limite.

Alocação de recursos para FL

Mecanismos de incentivo: incentivar a participação na federação de proprietários de dados com boa qualidade.

Propostas usam (não apenas):

- Teoria dos jogos: Jogo de Stackelberg;
- Teoria dos contratos.

Desafios:

- Clientes não compartilham informações sobre decisões:
  - Assim, usar uma forma fechada para definição de decisão é impossível;
- Difícil estabelecer a contribuição de cada participante para acurácia do modelo:
  - A acurácia do modelo depende da qualidade do dado e da complexidade do modelo.

Alocação de recursos para FL

A Learning-based Incentive Mechanism for Federated Learning:

- Servidor em nuvem publica uma tarefa para servidores em edge realizarem treinamento a partir de dados coletados de dispositivos IoT;
  - Servidor de parâmetros: - Despesa total;
  - Servidores edge: +Lucro = Recompensa do servidor de parâmetros - Custo de coleta dos dados.
- Usa jogo de stackelberg;
  - Modela o conflito.
- Usa aprendizado por reforço.
  - Para decisões não compartilhadas e avaliação de contribuição ambígua;
  - Aprende pelo histórico de registros de treino.

Alocação de recursos para FL

Modelo:

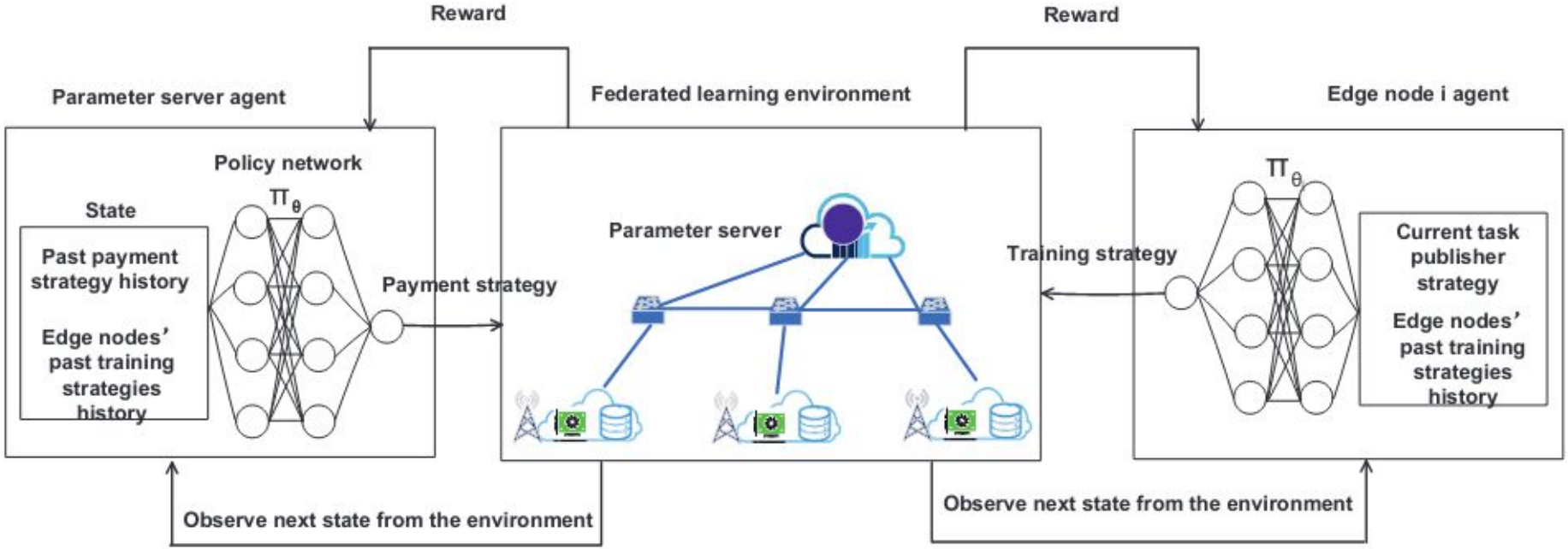
1. Servidor envia modelo anunciando pagamento total τ > 0;
2. Cliente decide participação com base nesse valor;
    a. Assumindo qualidade igual dos dados entre todos clientes e dados IID, a função utilidade é:

$$u_n(x_n, \boldsymbol{x}_{-n}) = \frac{x_n}{\sum_{m=1}^{N} x_m}\tau - c_n^{com}x_n - c_n^{cmp}x_n$$

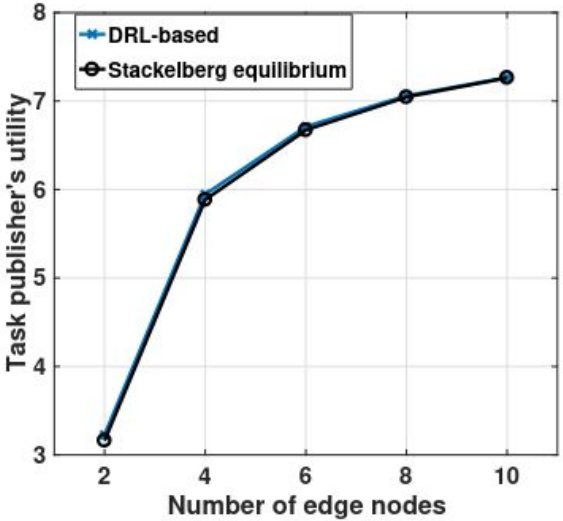
3. Servidor mede a utilidade da recompensa com:

$$u(\tau) = \lambda g(X) - \tau$$

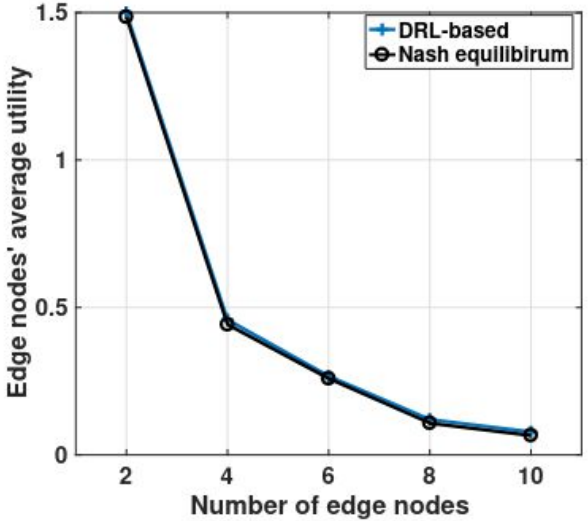
Alocação de recursos para FL



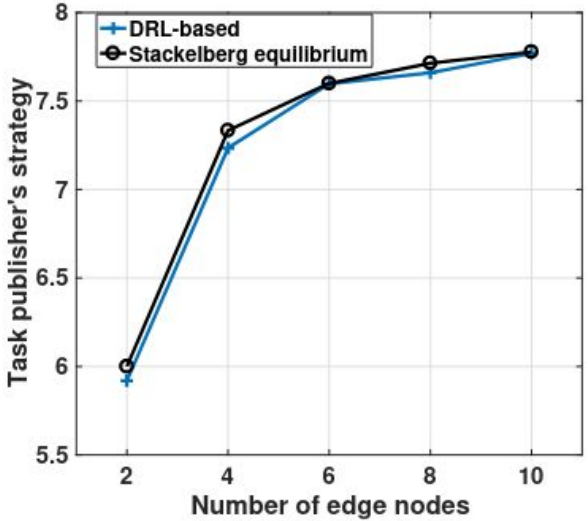
Alocação de recursos para FL



(a) Parameter server's utility.



(b) Edge nodes' average utility.



(c) Strategy of parameter server.

Alocação de recursos para FL

Resumo:

- A seleção de participantes pode otimizar o processo ao reduzir consumo desnecessário de computação e comunicação;
- FL síncrono depende do participante mais lento para agregação. FL assíncrono além de não ter esse problema, permitiria inserção no treino a qualquer instante;
  - A assincronia precisa melhorar porque não converge rápido.
- Mecanismos de incentivo são necessários para incentivar dispositivos a cooperarem com o treinamento e não apenas se beneficiarem do esforço coletivo realizado por outros.
  - Ajuda também a modelar o negócio, porque dispositivos podem ser de proprietários concorrentes.