AI for Mobile 5G:

Market-oriented Information and Resource Management

The story begins…

- Self-driving as an example
- Single / small-scale vehicular system
- Large-scale vehicular system
  - Fast and efficient data/information/service aggregation/request
  - Data driven local and global optimization
  - Data privacy versus data exchange
- What we do:
  - Distributed intelligence enabled mobile agents
  - Autonomous and self-governed mobile network for future mobile applications
  - Privacy and regulation concerns

Part I: Service-oriented communication systems: Softwarization and Learning-enabled 5G

Part II: Federated Learning for Mobile 5G Networks

Motivation and Challenges

- A large variety of applications with diverse requirements
  - higher data rates, reduced latency, enhanced system capacity, improved energy efficiency

- Network softwarization
  - Providing flexibility for mobile service management under dynamic network conditions and service demands
  - Supporting a large number of user equipments
  - Information/service centric, everything as a service

- Challenges in resource management in 5G
  - 5G network becomes increasingly heterogeneous and decentralized
  - Time-varying and highly unpredictable network environments (supply and demand)
  - All-in-one framework does not exist

Service and Resource Management in 5G

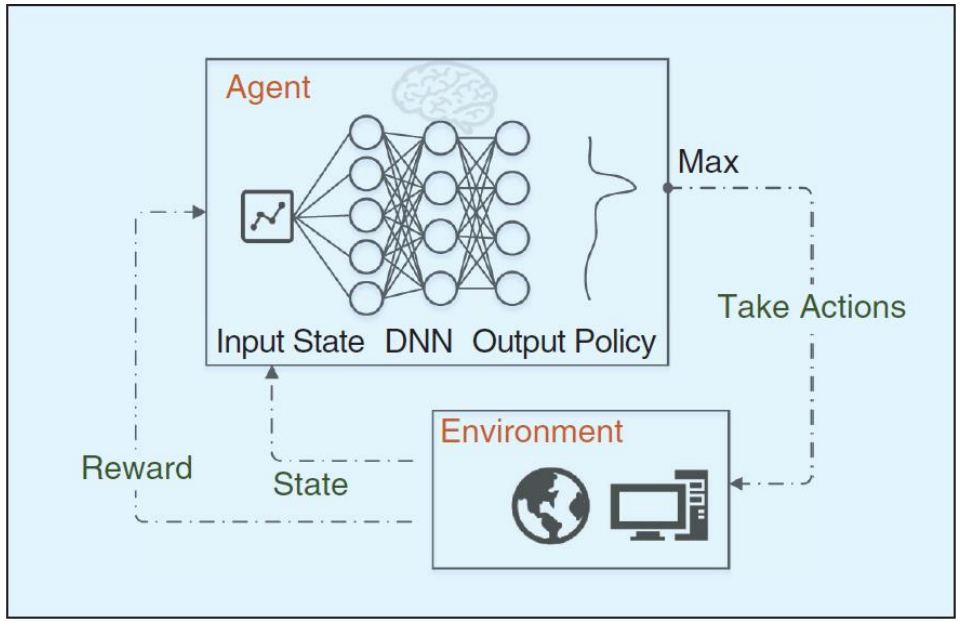
- Service/Resource Allocation in Communication Systems
  - Direct end-to-end service/resource delivery/management is not universally practical
  - Information-as-a-Service – Market-oriented comm. network information and service trading
  - Information vendor (Info-V) to distribute services to end users

- Economics of service transactions
  - Mutual influence and impacts among comm. system participants
  - Externalities:
    - Network effect
    - Congestion effect

Deep Reinforcement Learning: Intro

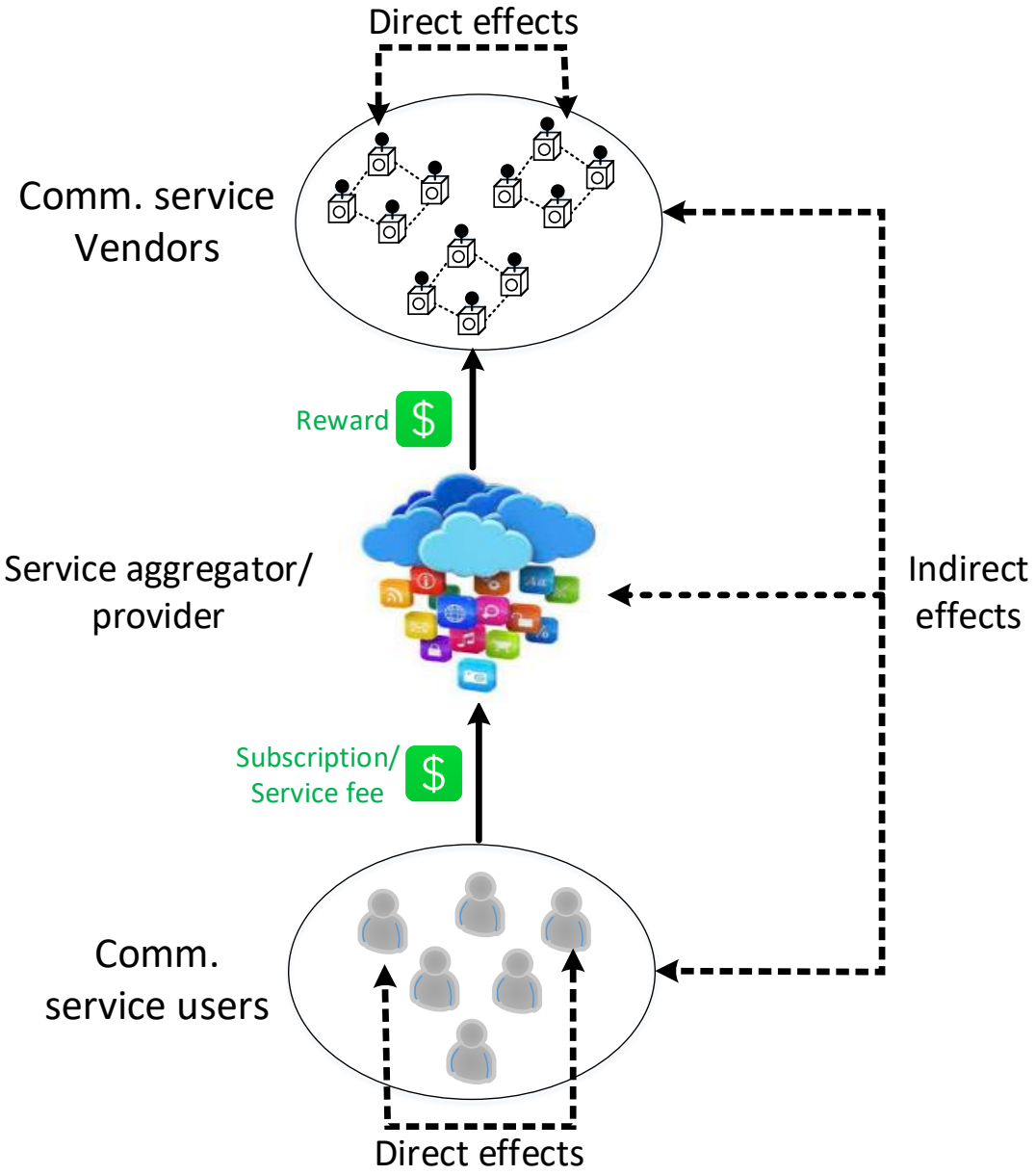
- Why learning is important in 5G
  - The curse of dimensionality
  - Network heterogeneity
  - From optimization to learning

- Deep reinforcement learning – a typical learning approach
  - Limitations in Solving Markov decision processes and reinforcement learning (RL)
  - Reduced model complexity in DRL - Tradeoff between imprecision and algorithm efficiency
  - Data driven - Evolved network performance over time



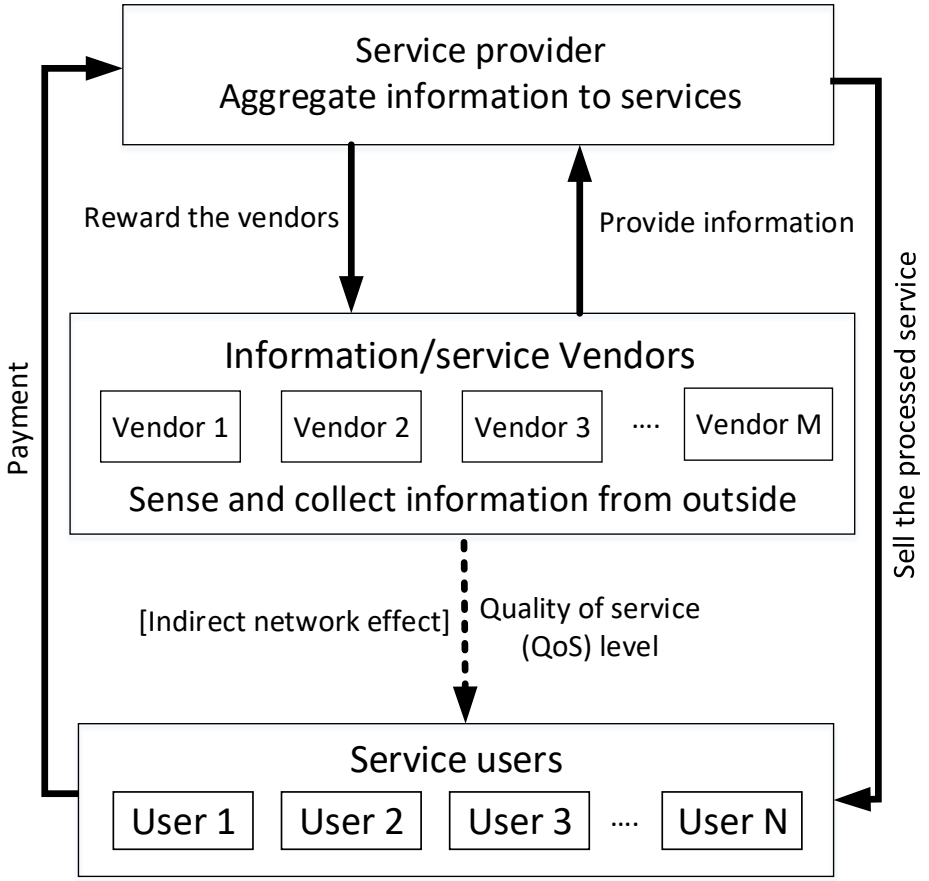
System Model Analysis: Components

- Service aggregator/provider
  - Works as a telecom SP / Edge server / BS
  - Providing IoT services to end users at subscription prices / negotiated prices
  - Prices decided by the SP
- Information/Service vendor (Info-V)
  - Owns IoT/sensing/service-related devices (e.g., nodes)
  - Providing raw data/services
  - For profit
- End user
  - Does not care about implementation details of services
  - Directly subscribes services from service aggregators/providers



System Model

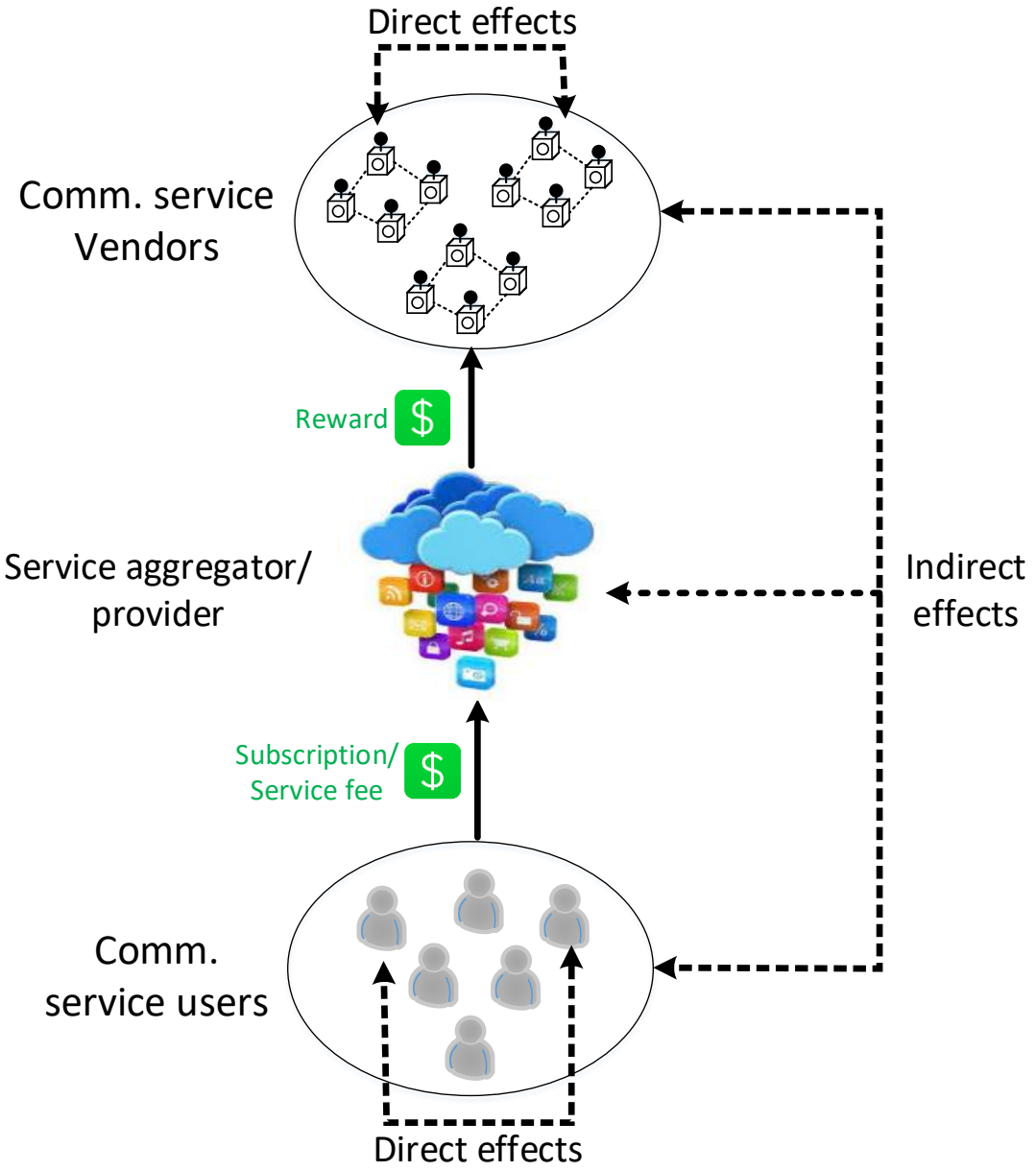
- From a user's perspective, s/he adopts a simple service request mode, that is, paying the SP and obtaining resources/information/services
- Info-Vs have reasonable information prices and low costs, but does not make transactions directly with end users. (e.g., like energy sector companies, power plants)
- SP, can be traditional telecom SP who provide services to users. Information requested by users can be prepared internally (by the SP), or the SP buys information from Info-Vs, and then sell the info as service bundles to users



Mutual Impacts among Network Components

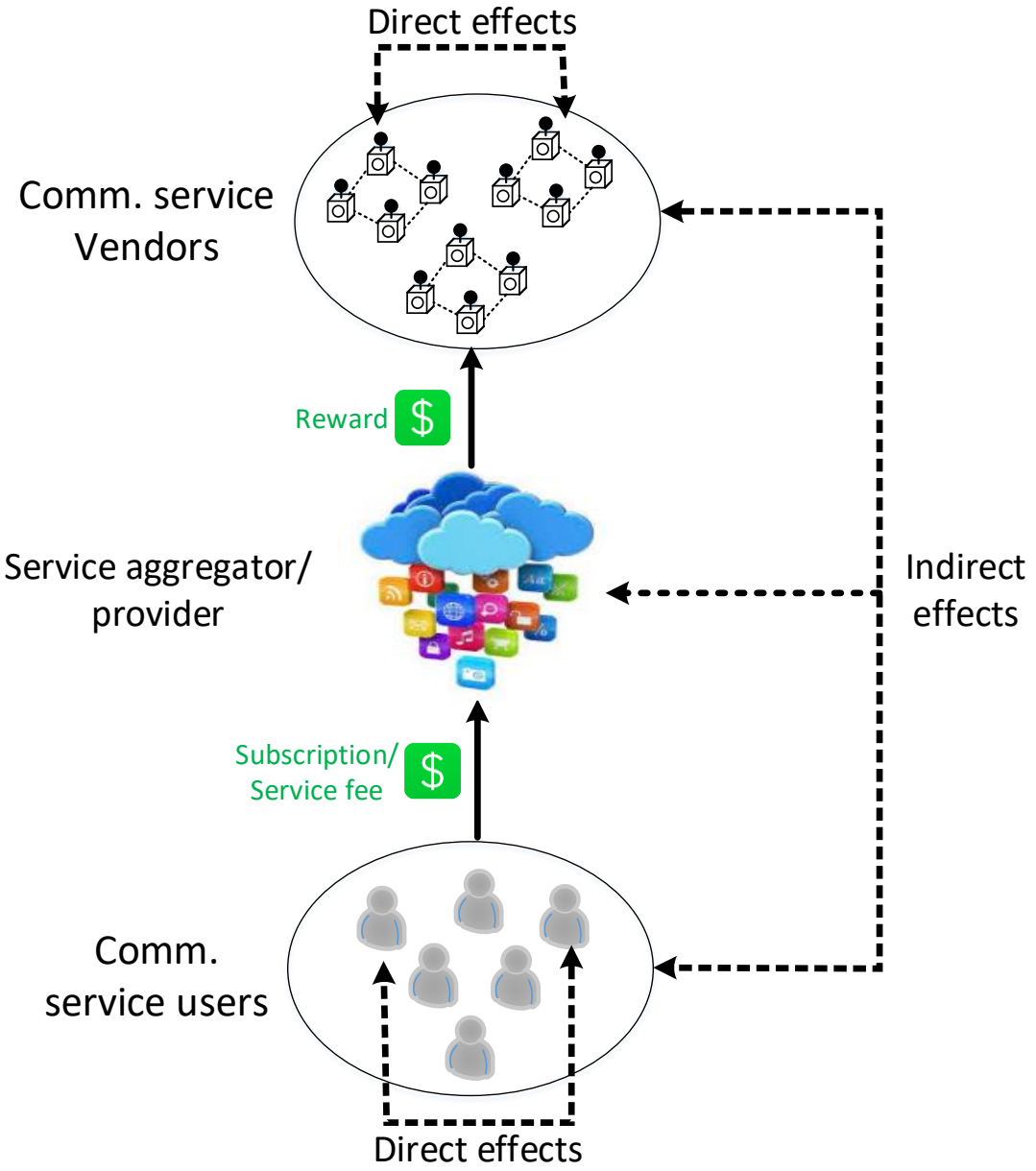
- Externalities
  - Effects brought by the existence of other system participants, e.g.
    - Network effect
    - Congestion effect

- External factors may have mixed effects to a user
  - Network and congestion effects can both exist at the same time.



Externalities and Market Factors: A User Perspective

- Internal effect
  - Diminishing returns, when buying services
- Indirect network effect
  - between Info-Vs and user: e.g., more Info-V, system is more attractive to the user
- Direct externalities from other users:
  - Network effect: e.g., more users, system is more popular to a incoming user
  - Congestion effect: e.g., more users, communication congestions happen



Stackelberg Game Formulation for Resource Allocation

- **Stage I (Service aggregator/provider)**: The SP acts as the role of leader in the proposed Stackelberg game, who decides the optimal service pricing plan p^*, as follows:

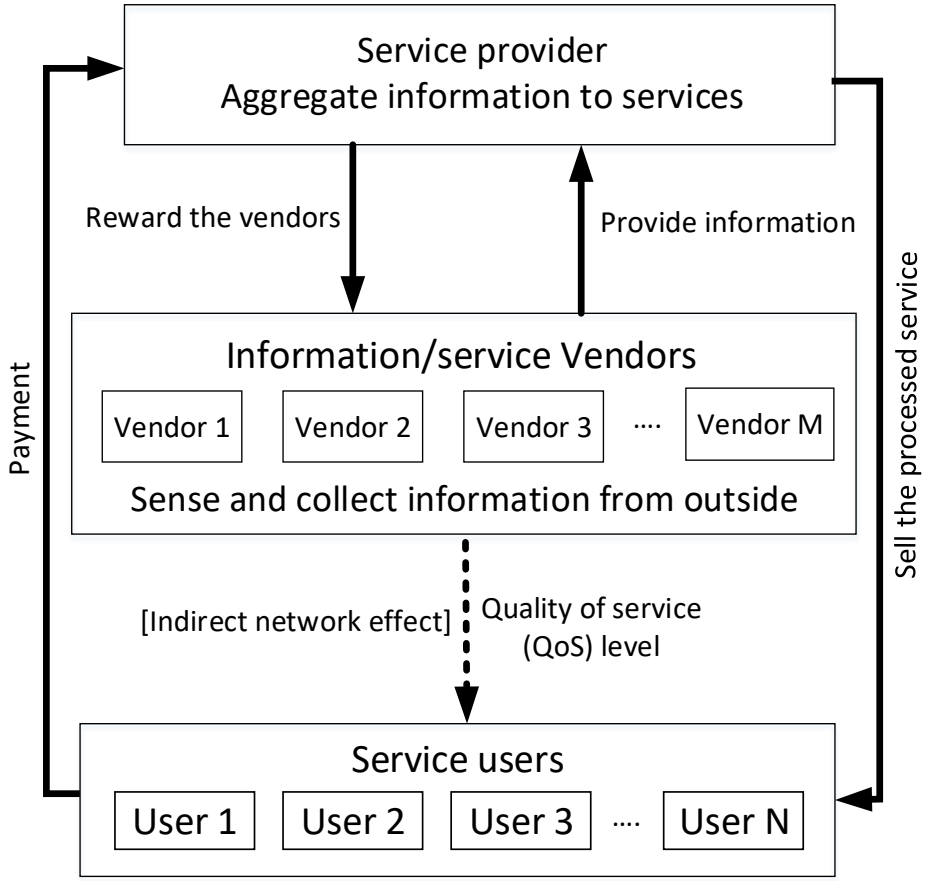
$$\text{Sub-game } \mathcal{G}_{SP}: \quad p^* = \underset{p \geq 0}{\operatorname{argmax}} \, \Pi.$$

- **Stage II (Information vendor)**: Given the optimal price p^*, each vendor j decides the reward r_j obtained from the SP to maximize the utility:

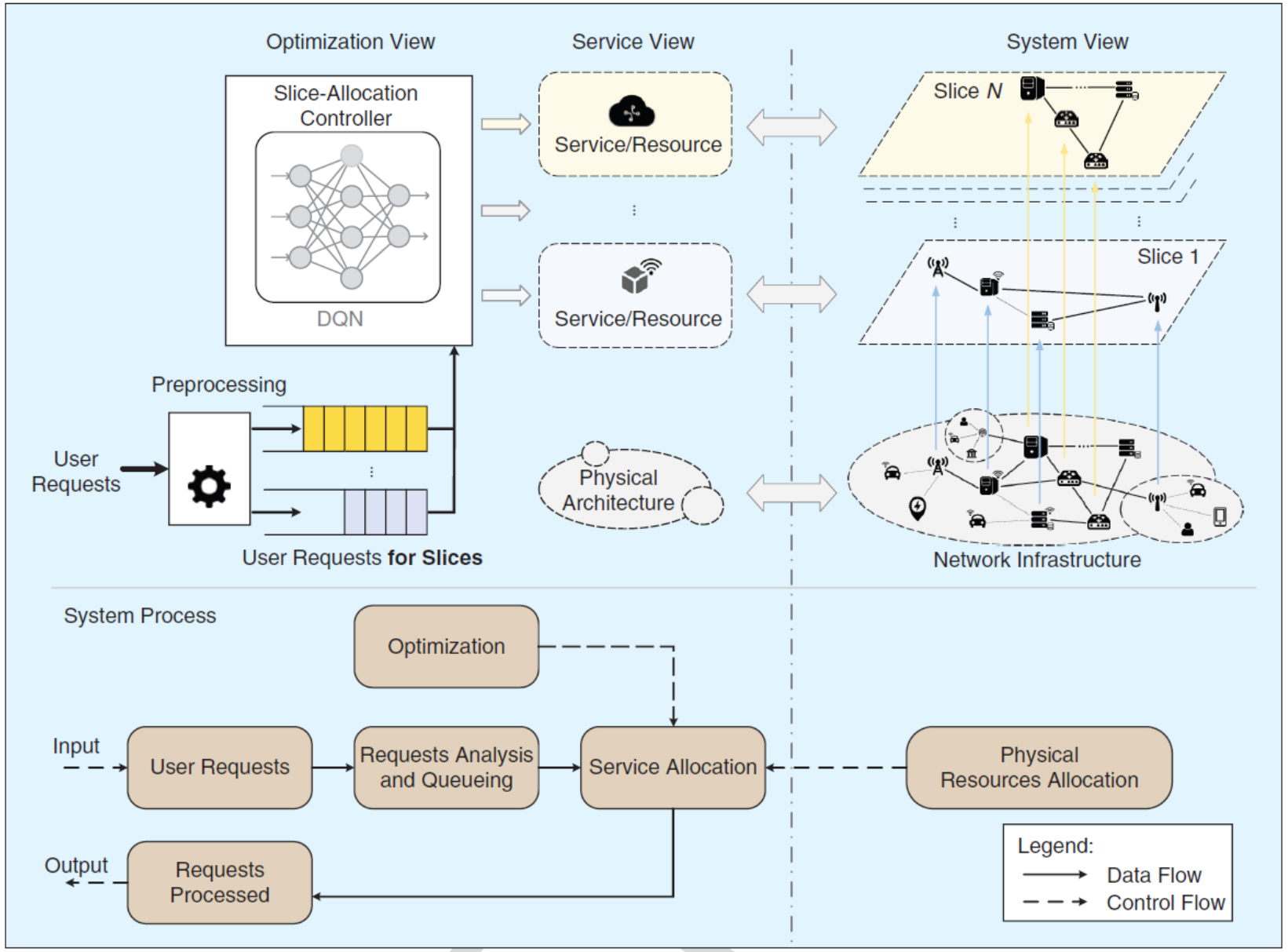
$$\text{Sub-game } \mathcal{G}_V: \quad r_j^* = \underset{r_j \geq c_j}{\operatorname{argmax}} \, \Gamma_j.$$

- **Stage III (Service user)**: Given the optimal price p^* and the optimal Info-V reward r_j^*, $\forall j \in \mathbb{M}$, each user determines the demand x_i^* to optimize its own utility, as follows:

$$\text{Sub-game } \mathcal{G}_U: \quad x_i^* = \underset{x_i \geq 0}{\operatorname{argmax}} \, u_i(x_i, \mathbf{x_{-i}}, \mathbf{\Gamma}, \mathcal{P}).$$



Network Slicing for Service Allocation: A DRL-based Architecture

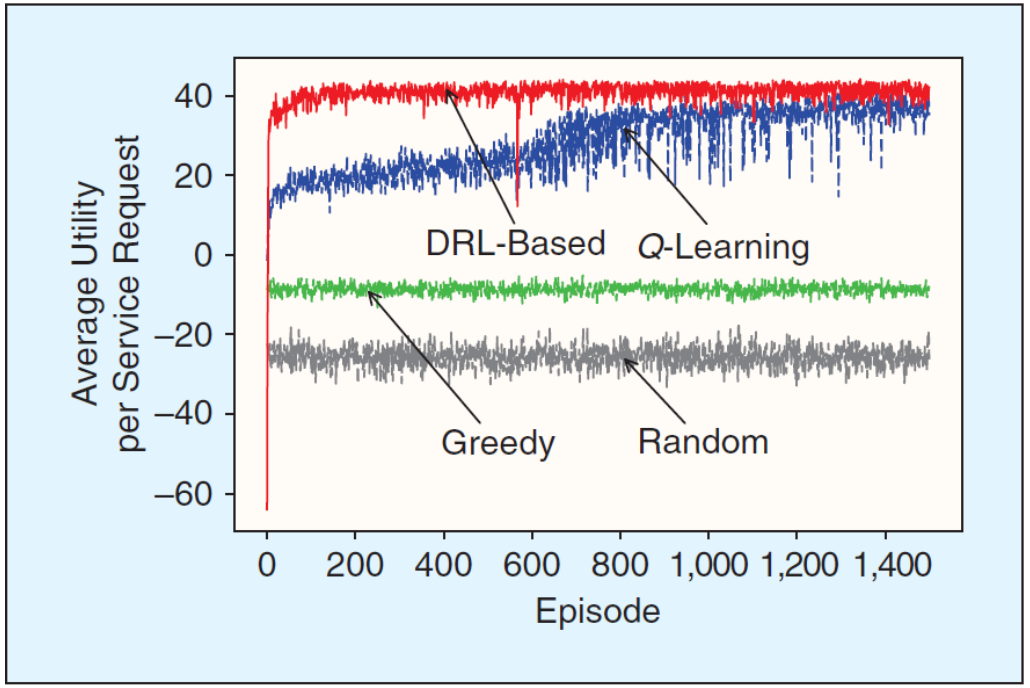


Network Slicing for Service Allocation: A DRL-based Architecture

- System perspective
  - Physical infrastructure virtualized into multiple service slices
  - Each slice provides a type of service
- Service perspective
  - Users request services instead of underlying infrastructure components
  - User requests to be well-handled
- Optimization perspective
  - A burst of user requests?
  - Queue-based model
  - Strategic slicing allocation to maximize user performance/service allocation efficiency/social welfare
  - RL/DRL based slice allocation

Results: DRL

- DRL model
- Observing system states
  - Slice queue length
  - Available resources in each slice
- State transactions
  - Allocated network slice consumes infrastructure resources
  - Releasing a network slice also releases corresponding resources
- Reward/token is recorded for every successful allocation
  - Maximize network slice utilization



Application Scenarios

- Power control and power management for physical layer
  - Cellular networks
  - Ultra dense networks
- Edge intelligence for supporting smart applications
  - Learning at the edge
  - Computing offloading
  - Edge caching for real-time services
- Network slicing for service-oriented mobile networks
  - Strategically organize networked resources and information
  - Providing services to fulfil diversified network requirements

Open Issues

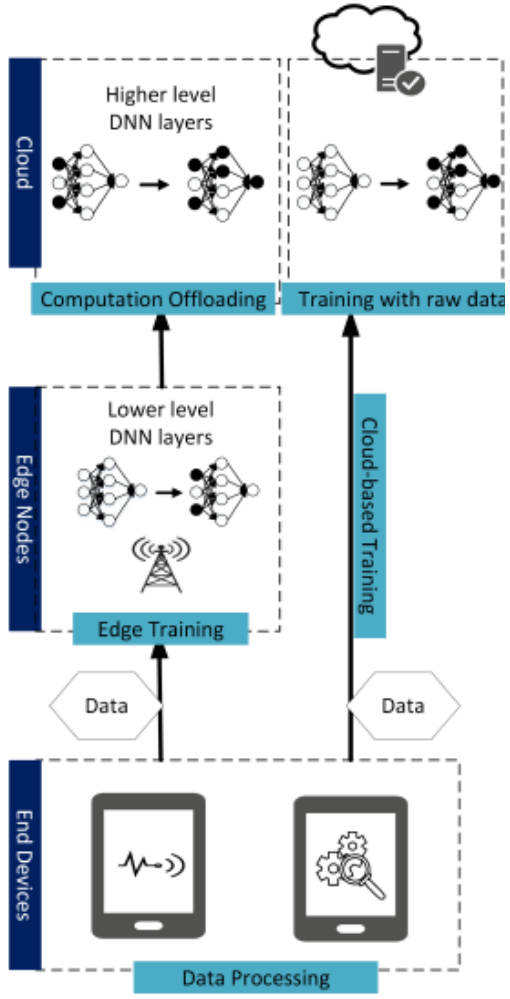
- Multi-agent learning in 5G
  - Phasing out centralized approaches
- Distributed learning and learning information transfer
  - Federated learning
  - Transfer learning
  - Information caching and privacy
- Measurement of social impacts and free-rider issues

Part I: Service-oriented communication systems: Softwarization and Learning-enabled 5G

Part II: Federated Learning for Mobile 5G Networks

Motivation

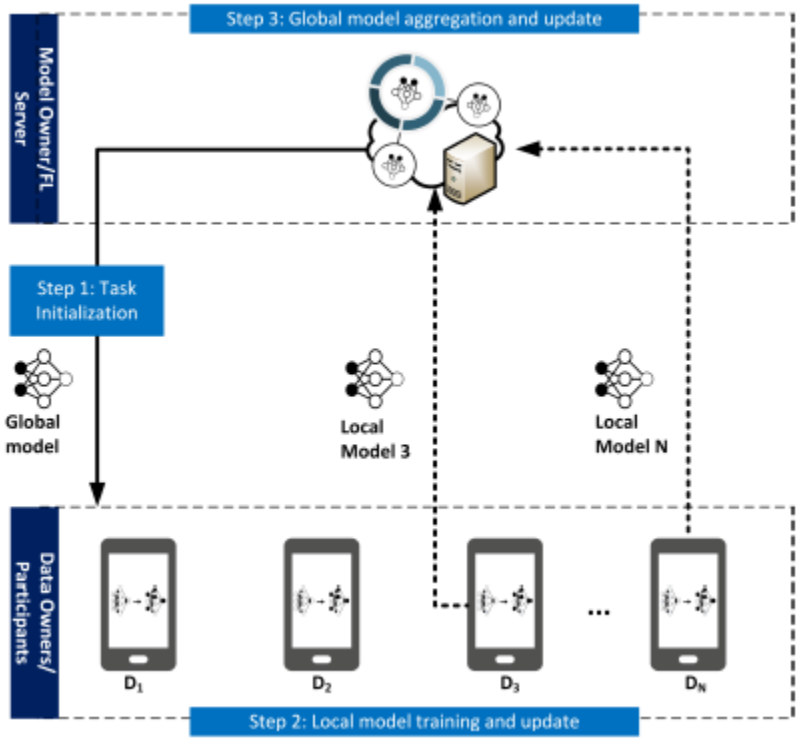
- IoT devices have increasingly advanced sensing/processing capabilities
- Useful for Deep Learning (DL) model training which require a lot of data
- Traditional cloud-centric and edge computing approaches require data uploading
- Need an alternative due to:
  - Data privacy concerns and increasingly stringent regulations
  - Intolerable latency for delay-sensitive tasks in mobile scenarios
  - Relieve burden on backbone networks especially for tasks that require constant model updates (esp. networks with increasing scale)



Federated Learning: Training Process

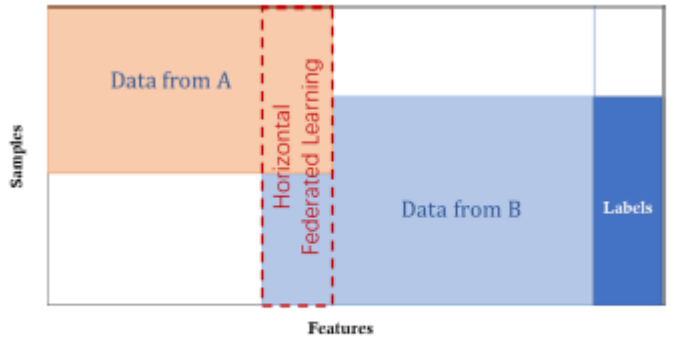
- For one iteration of the training process:

1. Participant selection
2. FL server sends global model and hyperparameters to selected participants
3. Participants train the model **locally** on their **local** data
4. Participants send the updated parameters or gradients back to the FL server
5. FL server takes a data quantity based **weighted** average across all received model parameters to update new global model

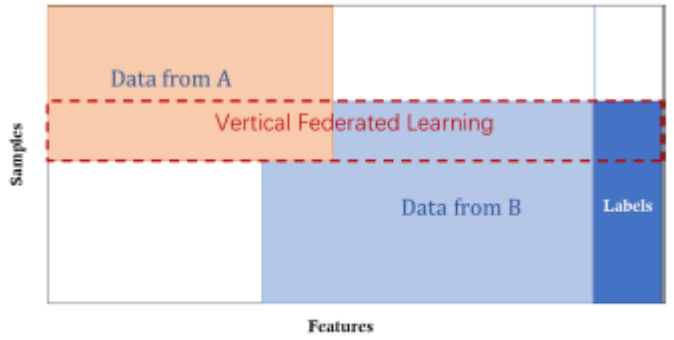


Categorizations of Federated Learning

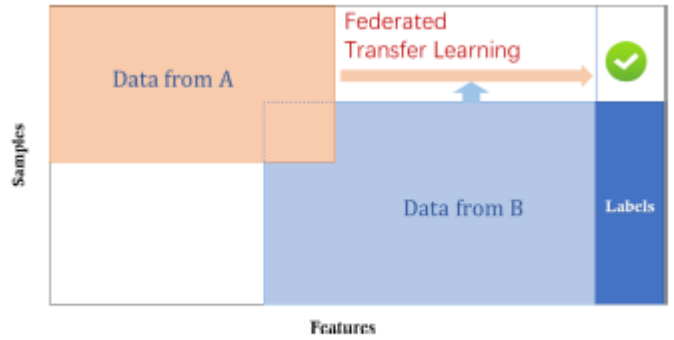
- Horizontal FL
  - Different sample IDs but with same feature space of interest
  - E.g. two banks with different user groups but similar feature spaces can work together
- Vertical FL
  - Different feature spaces but with same sample ID
  - E.g. banks with same users but each of which has data that capture different attributes of interest can work together
- Federated Transfer Learning
  - Different feature space and sample ID
  - E.g. when a commercial bank in China and e-commerce company in the US works together



(a) Horizontal Federated Learning



(b) Vertical Federated Learning



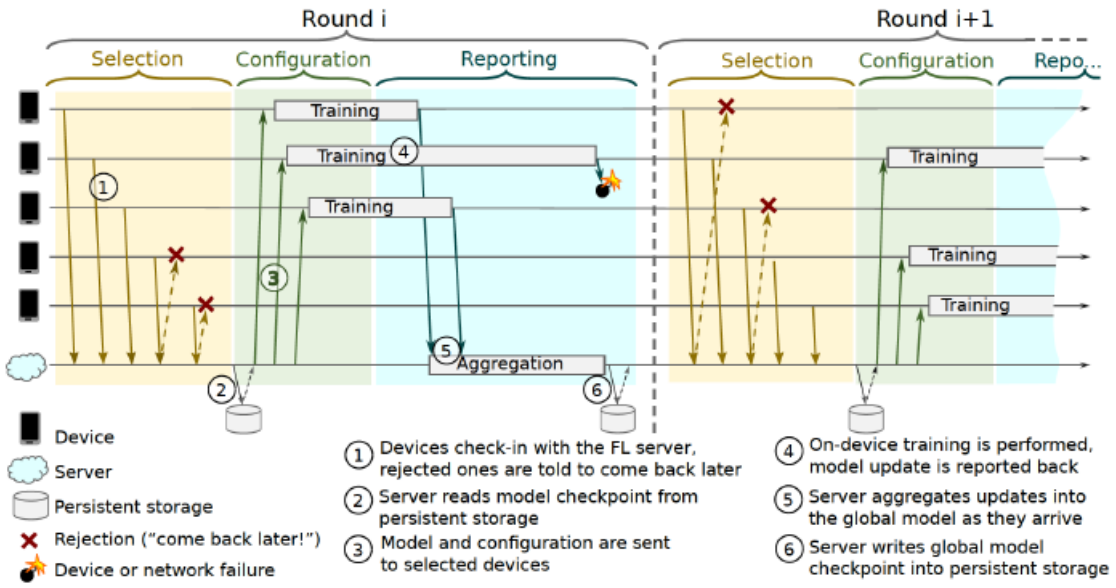
(c) Federated Transfer Learning

(Qiang, 2019)

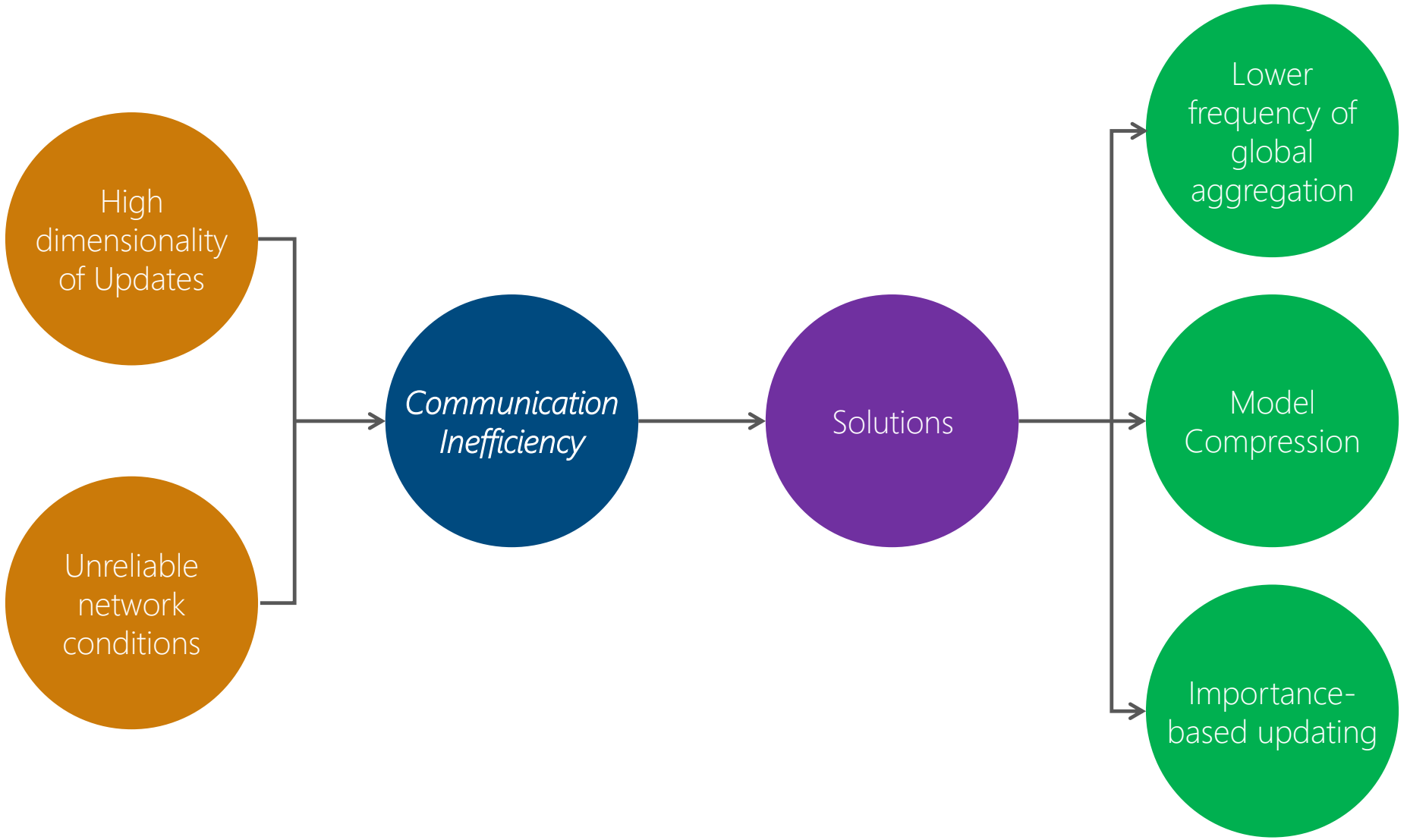
Federated Learning at Scale

- Three phases of the FL at scale protocol (Bonawitz, 2019):

1. Selection
   - Participant selection
   - Pace steering: Adaptively manage optimal time window for connection to FL server to prevent server overload/underload
2. Configuration
   - Server configuration of aggregation modes, e.g. secure aggregation and differential privacy, and training schedule
3. Reporting
   - Server receives updates from participants

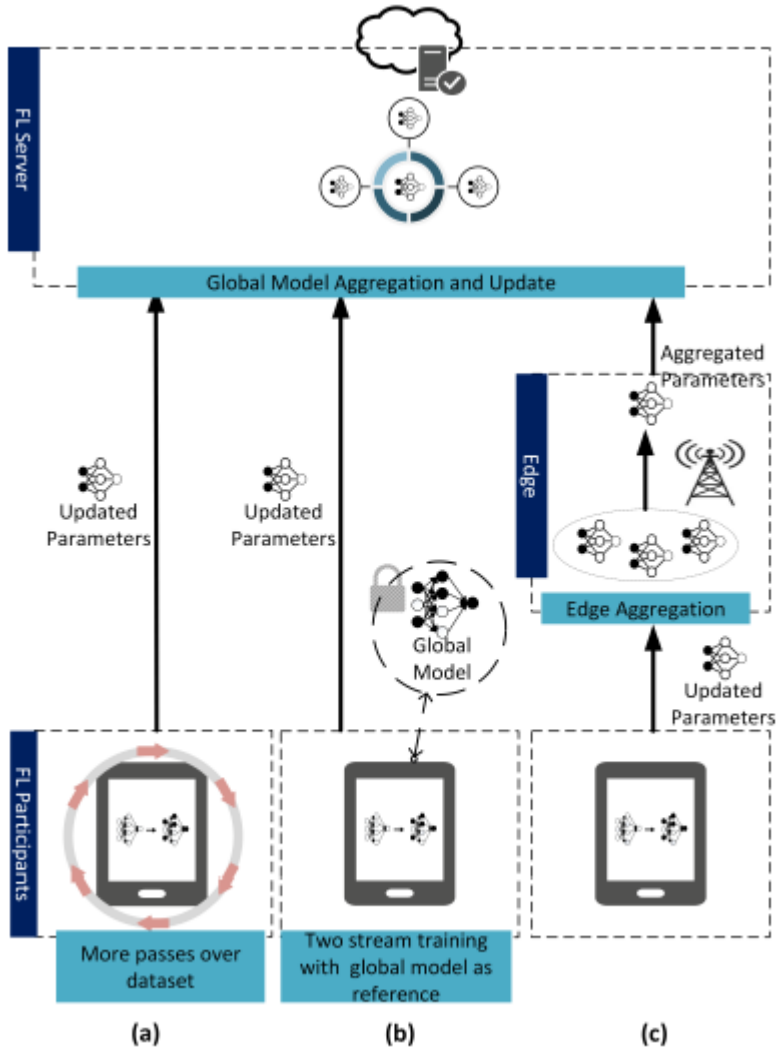


Issues in FL for 5G Networks: Communication Inefficiency



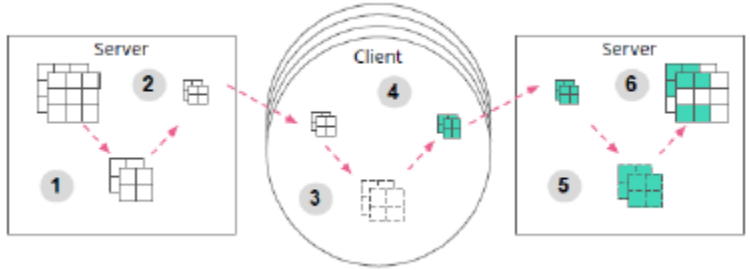
Lower frequency of Global Aggregation

- Computation is not the only bottleneck; *communication also is*
- Increase computation rounds at end devices and reduce global aggregation frequency to lower communication costs
- Existing studies:
  a) Participating devices can make more passes through dataset or use smaller minibatch size (McMahan, 2016)
  b) Two-stream FL: Local training on each device to minimize maximum mean discrepancy relative to global model, so as to reduce communication rounds (Yao, 2018)
  c) Device communicates updates to an intermediate edge server first rather than directly with remote cloud. After some iteration, edge server then communicates with cloud for aggregation (Liu, 2019)



Model Compression

- Neural network models can involve millions of parameters
- Reduce the size of high dimensional models for communication efficiency
- In particular, model uploading is more inefficient than downloading **due to slower upload speed**



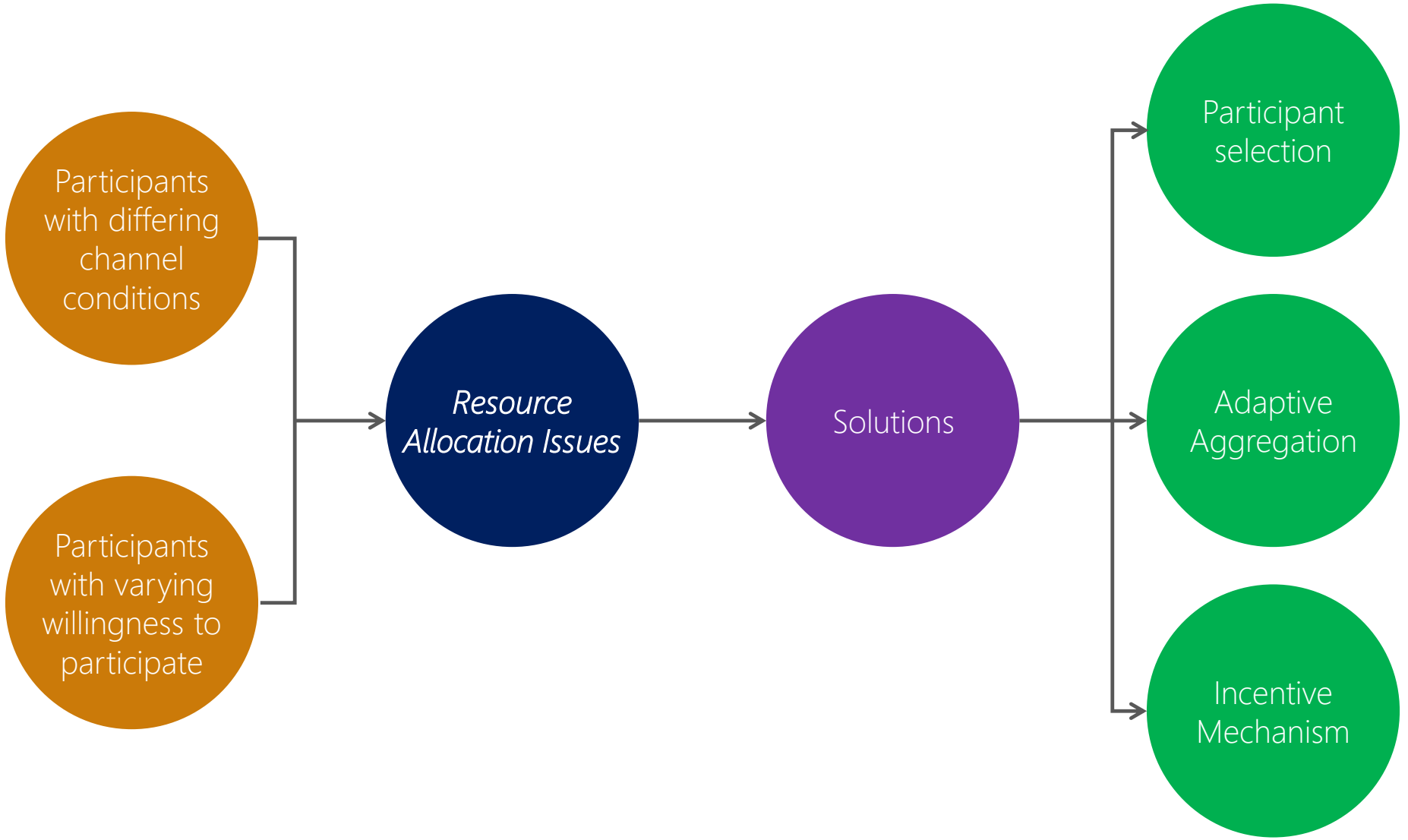
- Caldas (2018) also considered compression for model downloading
- Solutions (Caldas, 2018):
  1. Federated dropout: Remove some activation functions to derive a smaller sub-model before sending the model to participants
  2. Compression of model before server communicates with client

- However, compression comes at the expense of **reduced accuracy** and sometimes convergence delays
  - Especially when dropout rates are high

Importance-based Updating

- Most parameters of models have weights that are close to zero
- Select only small fraction of parameters to be communicated with FL server
- Examples of existing works
  - Selectively communicate parameters that reduce the training loss (Tao, 2018)
  - Compute relevance score based on sign of parameters between local and global parameters of previous iteration. Irrelevant local updates are dropped (Wang, 2019)
- However, the dropping of parameters can harm model accuracy

Issues in FL for 5G Networks: Resource Allocation



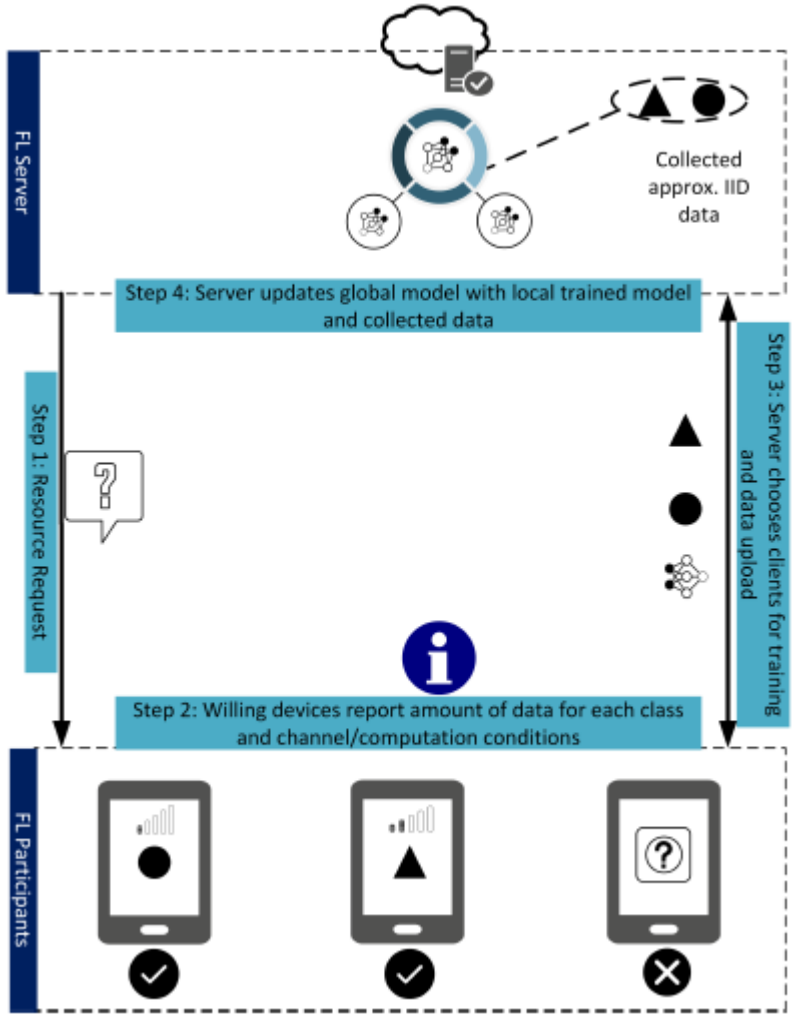
Participant Selection and Adaptive Aggregation

- Straggler's effect:
  - Global model aggregation **may** only take place when **all** local model updates are received
  - FL training round takes as long as the slowest device

- Greedy algorithm to select maximum possible number of participants that can complete training within pre-specified deadline (Nishio, 2019)

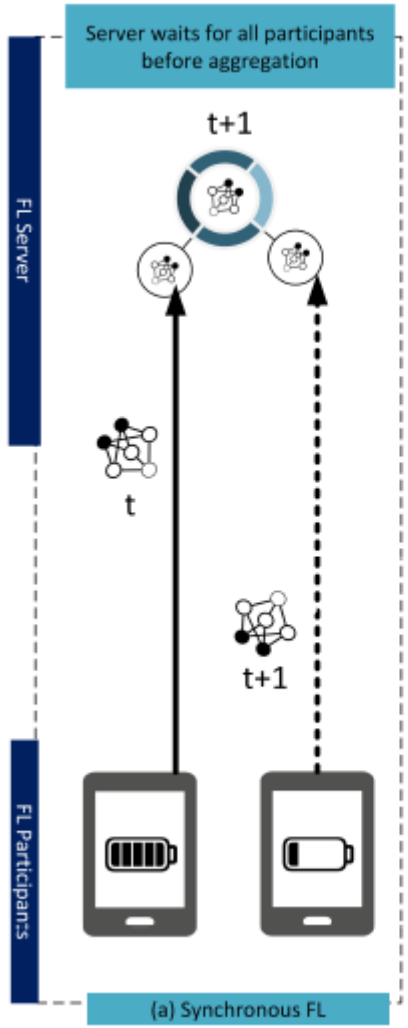
- Also, select devices that are willing to upload approximately-IID data

- Deep Reinforcement Learning (DRL) approach to decide amount of data, energy, CPU resources each mobile device uses to reduce energy consumption and training latency (Anh, 2019)



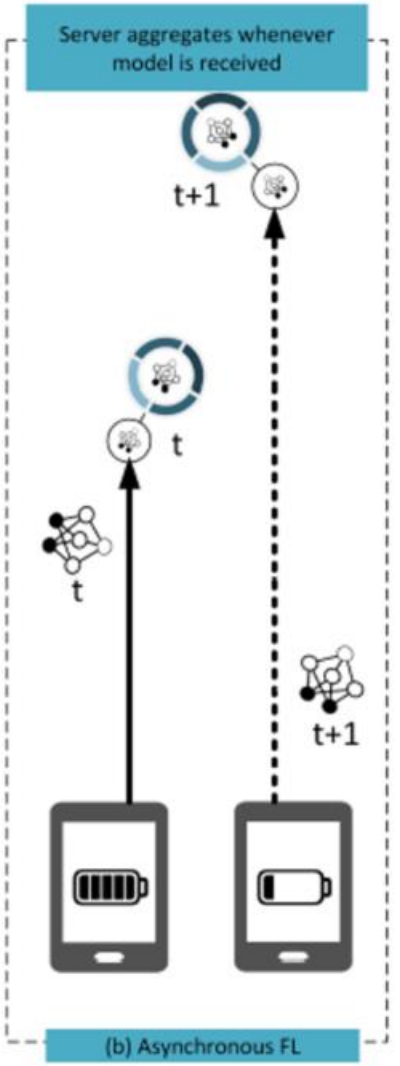
Participant Selection and Adaptive Aggregation

- Previous studies may have considered FL using synchronous FL
    - All local model parameter updates have to arrive before server updates global model
- However, this leads to straggler's effect → each training iteration is as slow as the slowest device
- Participants joining halfway when the training is already taking place are also left out



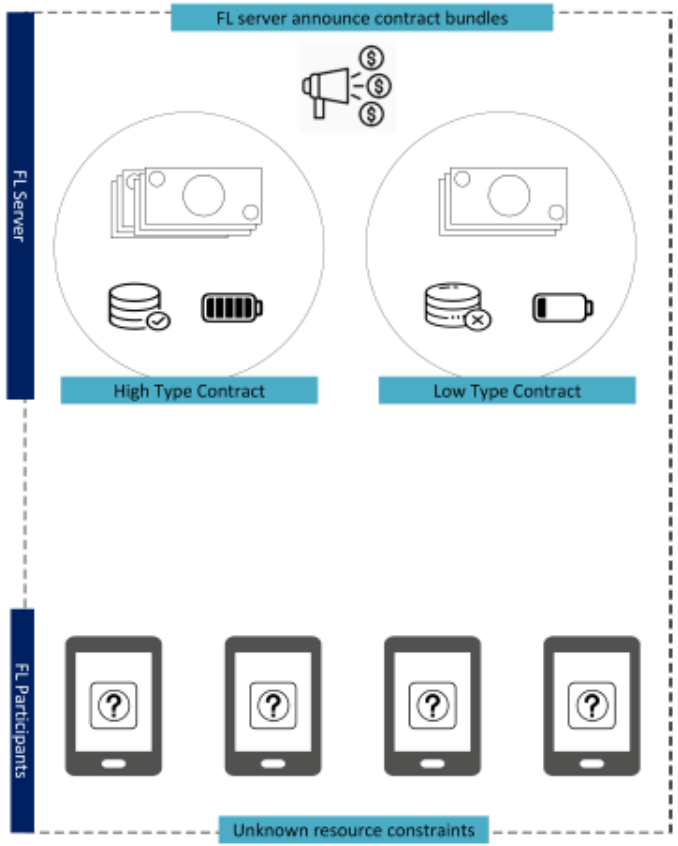
Participant Selection and Adaptive Aggregation

- Possible solution: Asynchronous FL
    - Server updates global model whenever it receives a local update
- Robust to participants joining halfway during a training round (Sprague, 2018)
- Newly received updates are weighted according to "staleness", i.e., iteration it was updated vs current iteration. A stale update is weighted less (Xie, 2019)
- Control algorithm to adaptively choose global aggregation frequency based on system states, e.g., wireless network conditions (Wang, 2019)
- However, asynchronous FL has convergence issues for non-IID and unbalanced data (Sprague, 2018)
- Also, straggling devices can be underrepresented in the model (Li, 2019)



Incentive Mechanism

- Participants have heterogeneous types, e.g,:
  - Levels of willingness to participate
  - Quality and quantity of data
  - Computing resources
- Also, information asymmetry exists:
  - Adverse selection: Undesirable participants lie about their types to be selected/well rewarded for FL training
  - Moral hazard: Participants put in less effort after being selected for FL training
- Solution: Incentive Mechanism Design



Incentive Mechanism

- Stackelberg game in monopoly markets
  - Upper level: Model owner chooses optimal compensation scheme
  - Lower level: Participant chooses optimum effort in response to model owner, e.g., compute resource
- Cooperative relay network to support model update transfer and trading (Feng, 2018)
- Incentivize participants to contribute more computation power (Sarikaya, 2019)

Incentive Mechanism

- Contract theory (Bolton, 2005)
  - Self-revealing mechanism
  - Individual rationality: Participating in FL at least yields a non-negative utility
  - Incentive compatibility: Participants only choose the contract designed for their types
- Contract theory to incentivize high quality data contributions (Kang, 2019)
- Select participants for each training round based on reputation (stored on blockchain)
  - Reputation is updated after each round to reduce moral hazard

Application Scenario: Edge Caching and Computation Offloading

- Edge servers have computation and storage capacity constraints
  - Some tasks have to be offloaded to remote cloud
- FL approach to optimize caching and computation offloading decision while preserving privacy:
  - Federated DRL to optimize decisions under different states e.g. wireless network conditions, task queuing (Wang, 2018)
  - Federated DRL in IoT systems (Ren, 2019)
  - FL based stacked autoencoder learning model to predict content popularity for caching (Yu, 2018)
  - Privacy-aware service placement scheme (Qian, 2019)

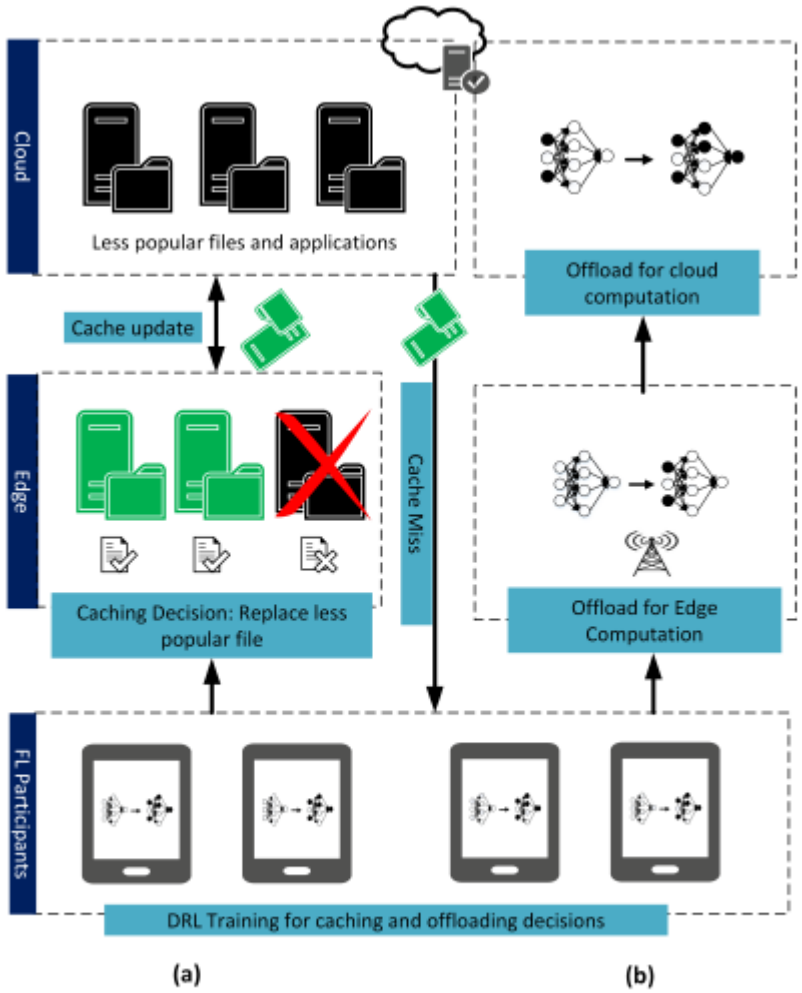


Fig. 16: FL-based (a) caching and (b) computation offloading.

Application Scenario: Vehicular Networks

- Radio resource management techniques need to account for extreme events to enable Ultra reliable low latency communication (URLLC) for intelligent transport system
  - An asynchronous FL approach to train local models on individual vehicle's queue state information (Samarakoon, 2018)
  - Roadside units update the global model via aggregation of local models, before returning the updated global model to the vehicular users
  - Queue state information **remains on each vehicle**, only the model parameters are exchanged → Privacy preserving

- Electric vehicle charging can lead to energy transfer congestion when too many vehicles congregate for charging
  - Federated Energy Demand Learning (FEDL) to forecast energy demand for electric vehicular network (Saputra, 2019)
  - Data is **kept** separately **at charging stations**, thus ensuring privacy preservation

Thank You.

yangzhang@whut.edu.cn
opt-yang.github.io