Élő dolgok Internete (Internet of Living Things)
GINOP-2.3.2-15-2016-00037

SZÉCHENYI 2020

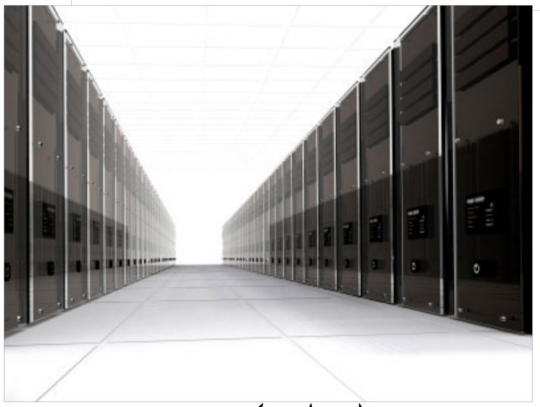
Federated Learning on the Edge

Árpád Berta, Gábor Danner, István Hegedűs,
Márk Jelasity,
(University of Szeged, Hungary)

User privacy is an increasingly important concern

- Personal devices (phones, tablets, IoT devices, etc.) are a rich source of user-data
  - Interactions with apps, behavior, etc
- This data is highly sensitive, collecting it will become increasingly hard
  - EU: General data protection regulation (GDPR)
  - US: Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy (White House Report, 2012)
- Let us design algorithms that leave raw data in place!

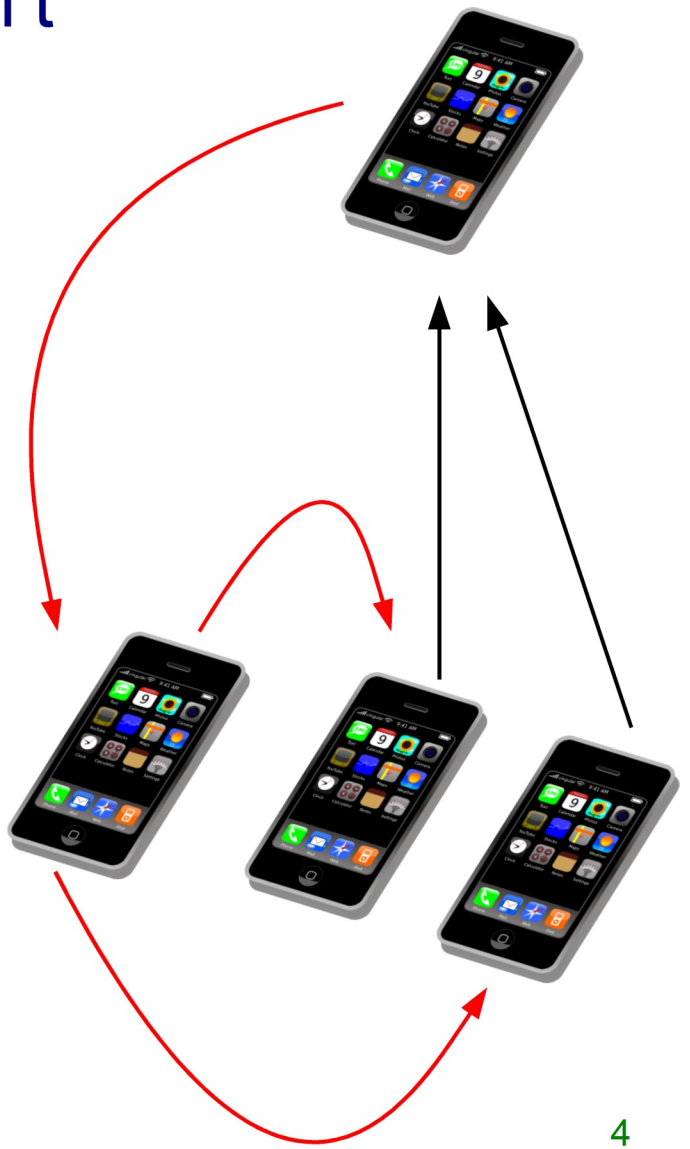
Federated learning (Google)



- Phones collect data locally
- Learning algorithm
  - The phones
    - **Update current model based on local data**
    - **Send the model to the server**
  - The server
    - **Averages the models**
    - **Sends back the new model**
  - Repeat
- Like the parameter server but with rather different assumptions

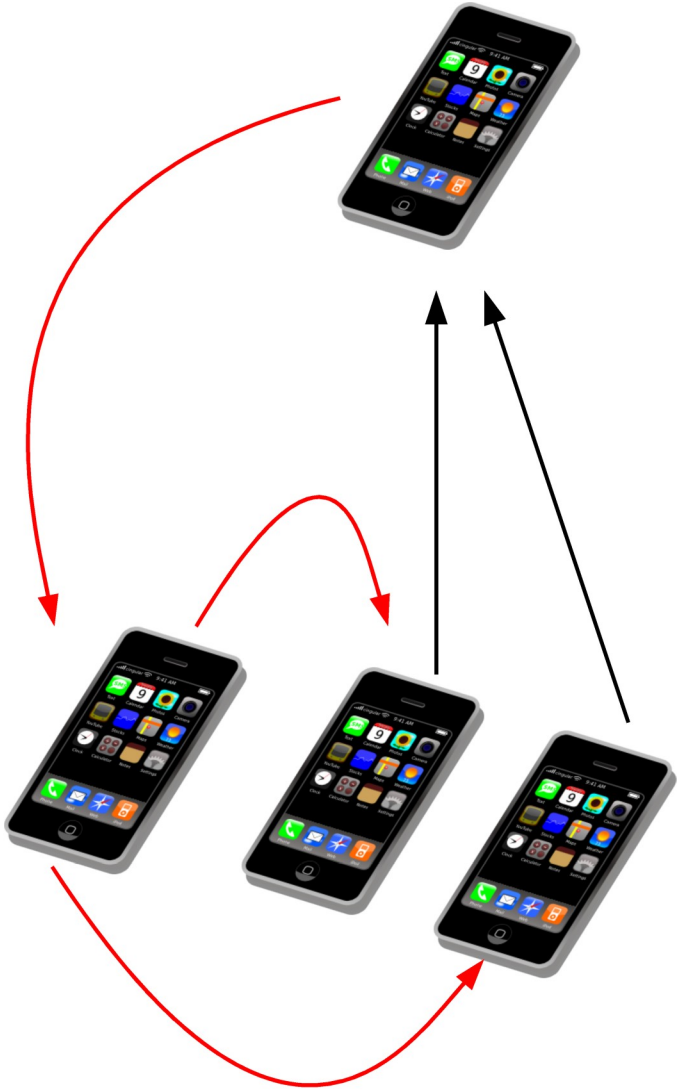
Gossip learning

- If data can stay in the edge, why can't communication stay too?
- In fact gossip learning implements exactly that
- Possible advantages include
  - No single point of failure
  - Very low cost of entry
  - Unlimited scalability
  - Potentially more privacy (if done right)
  - Independence of infrastructure, censorship, financial interests, and so on



Gossip learning

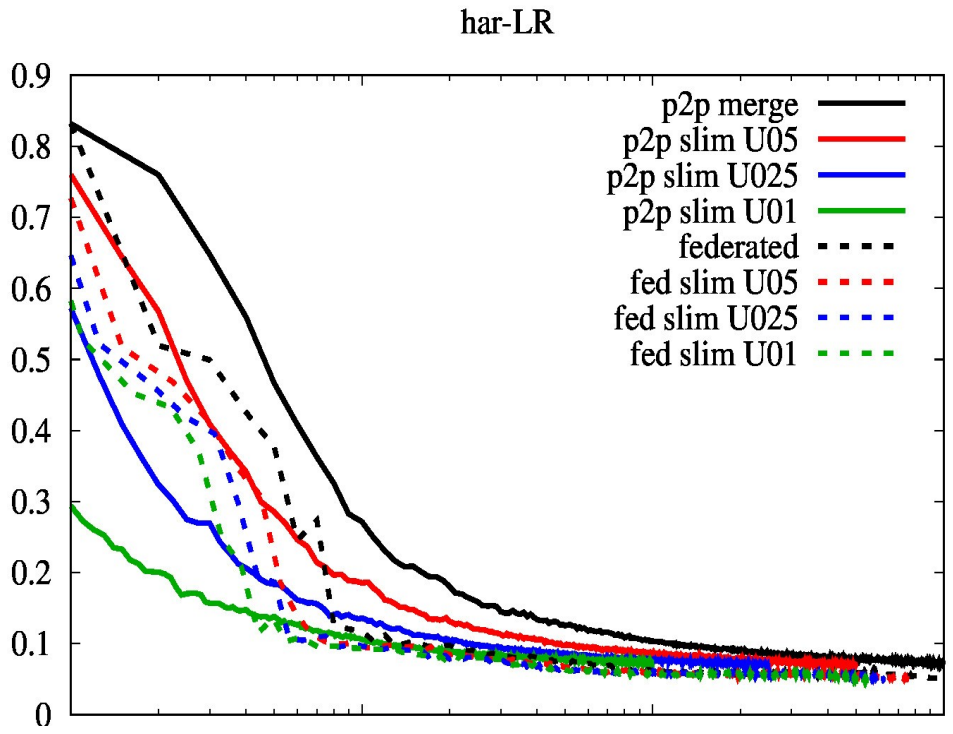
- Phones collect data locally
- Learning algorithm
  - Every phone
    - **Updates its current model based on local data**
    - **Sends model to a random peer**
    - **In the meantime merge (average) incoming models to current model**
  - Repeat
- Lots of nitty gritty details
  - Handling churn, learning rate, concept drift, etc.

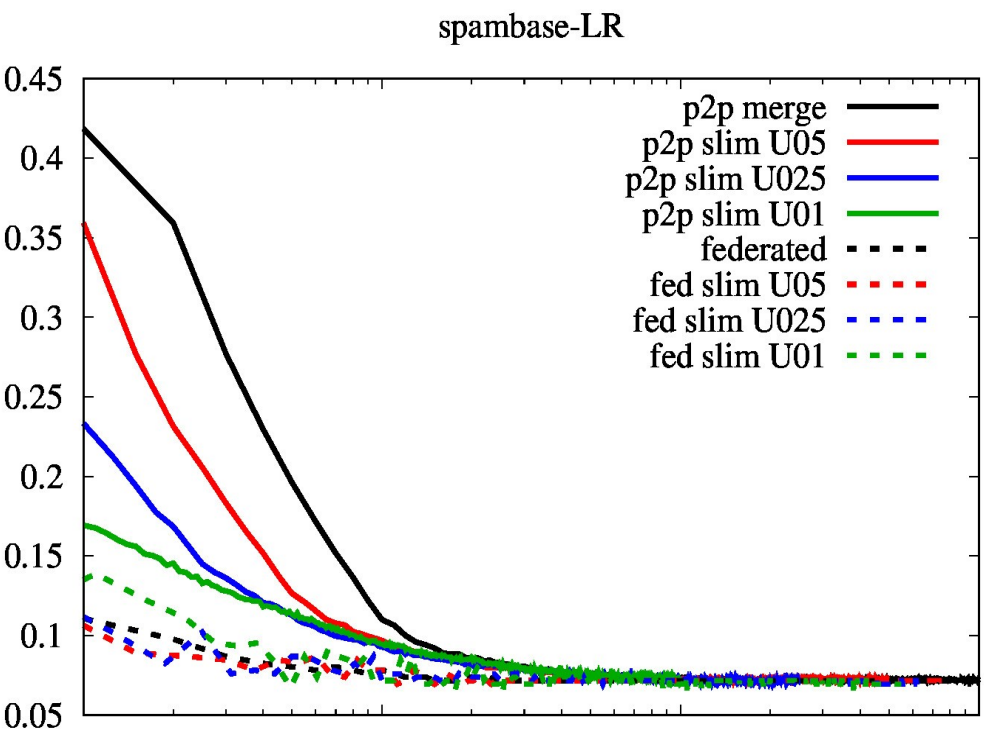


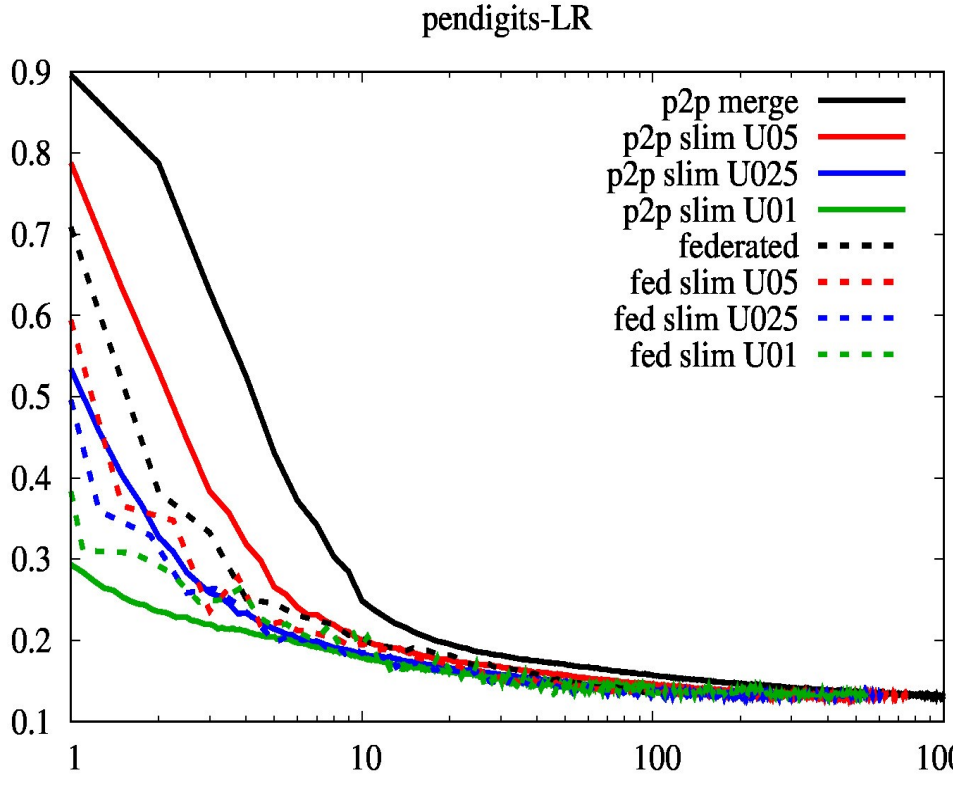
Experimental results

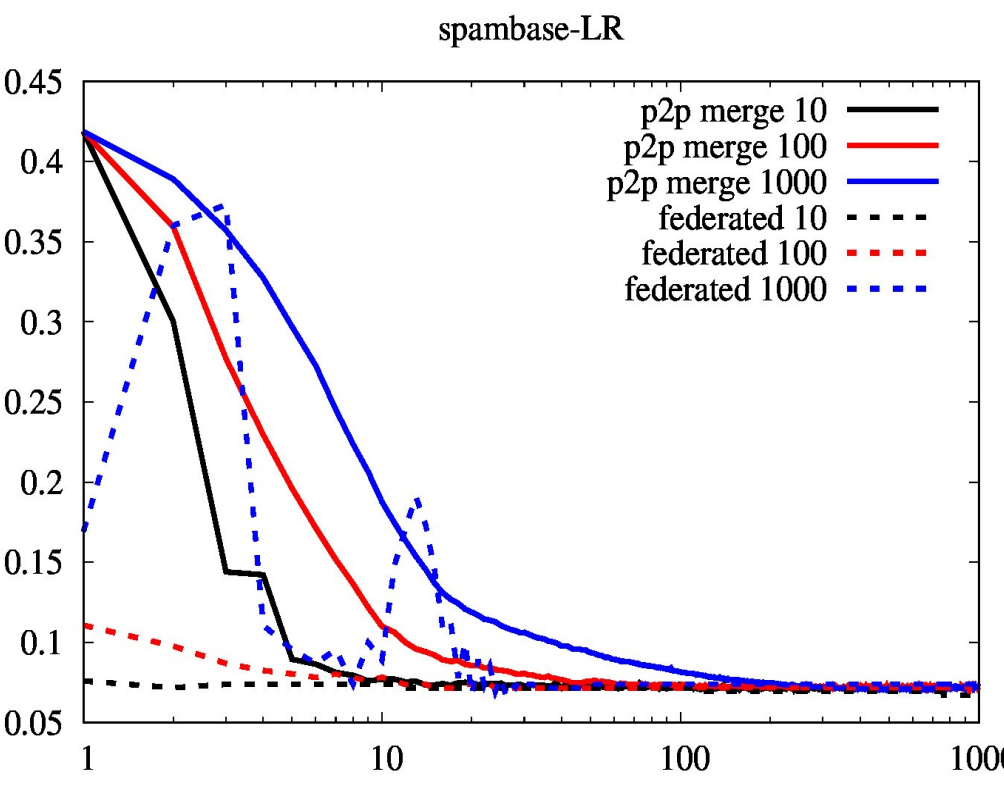
- 100 nodes
- Examples distributed uniformly
- Locally
  - 1 epoch
  - batch size: 10
- Logistic regression
- Uniform parameter sampling (100,50,25,10%)

- SpamBase
  - 57 features, 2 classes, 4140/460 (train/test)
- PenDigits
  - 16, 10, 7494/3498
- Human Activity
  - 561, 6, 7352/2947

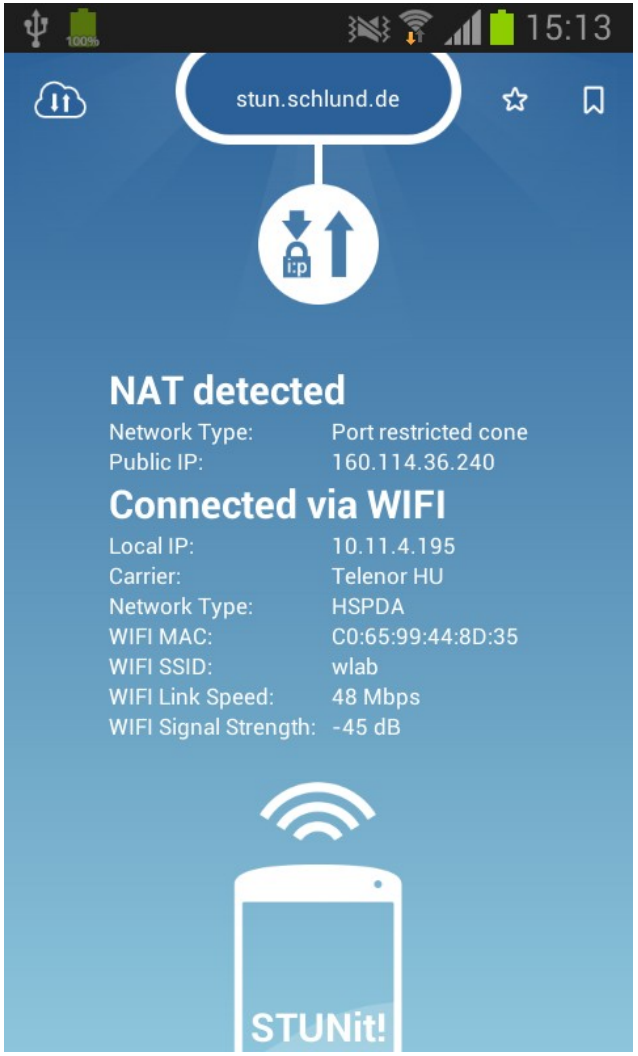

har-LR
p2p merge
p2p slim U05
p2p slim U025
p2p slim U01
federated
fed slim U05
fed slim U025
fed slim U01
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


spambase-LR
p2p merge
p2p slim U05
p2p slim U025
p2p slim U01
federated
fed slim U05
fed slim U025
fed slim U01
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05

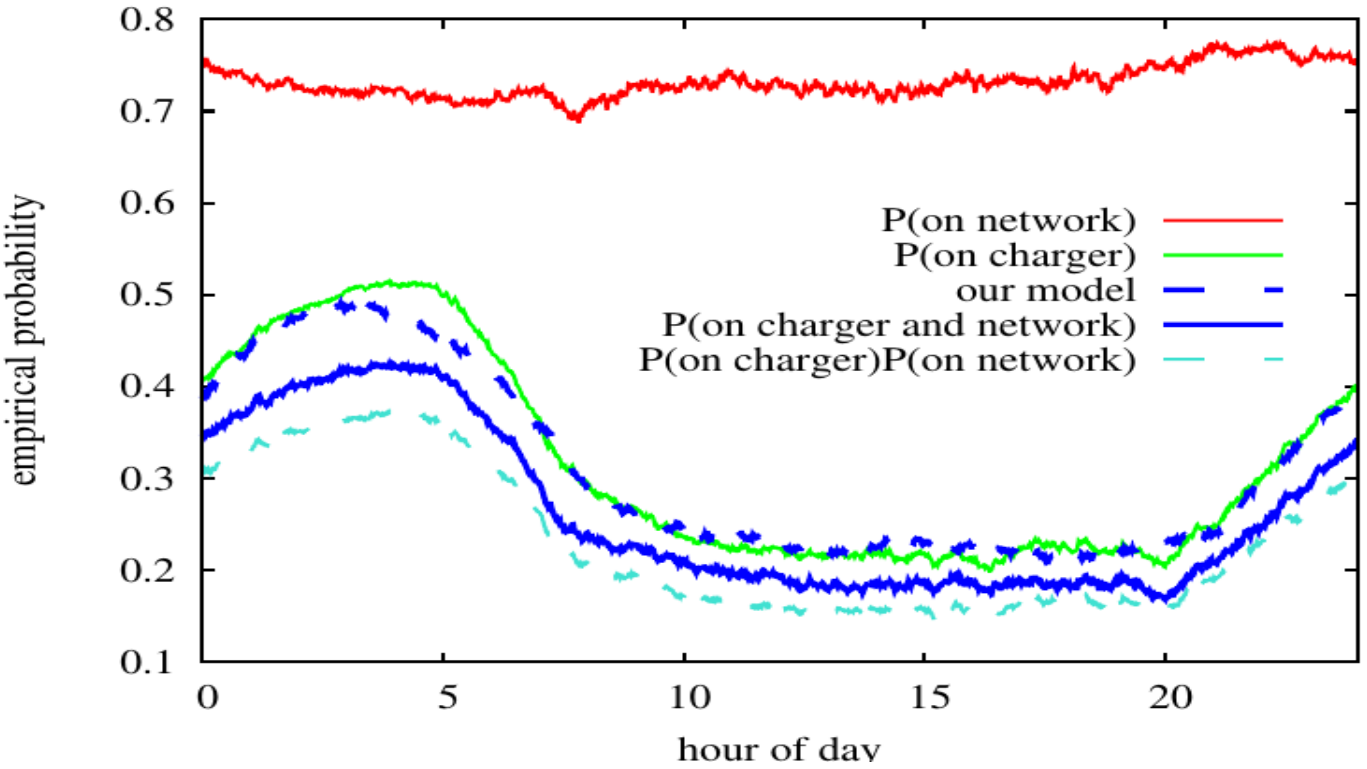

pendigits-LR
p2p merge
p2p slim U05
p2p slim U025
p2p slim U01
federated
fed slim U05
fed slim U025
fed slim U01
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1
10
100


spambase-LR
p2p merge 10
p2p merge 100
p2p merge 1000
federated 10
federated 100
federated 1000
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
1
10
100
1000

Data collection for simulations



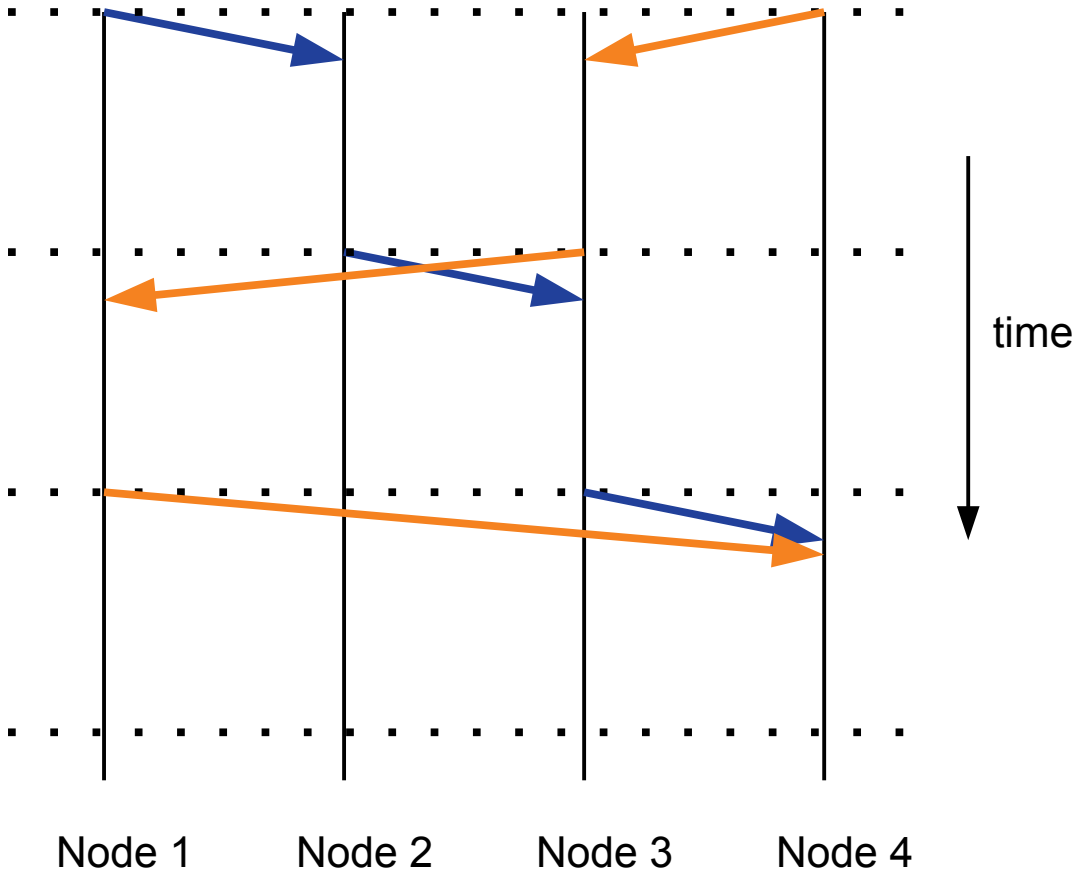
- Stunner application (Android)
- Has been running for years
- New completely rewritten release will be out soon with P2P measurements



Additional improvements

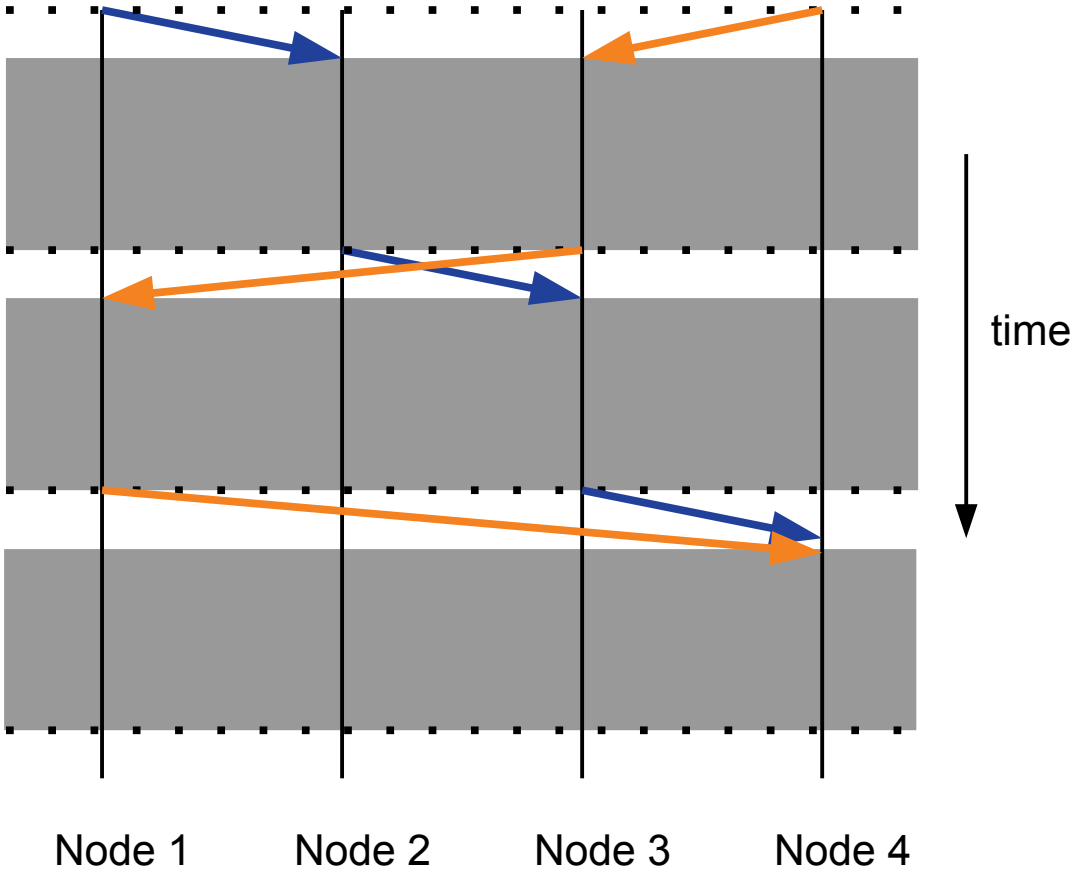
- Compression algorithms
  - Several approaches (sampling, quantization, stateful compression codecs, etc)
- Security approaches
  - Differential privacy, secure computation of mini batches, and so on…
- Flow control with the token account algorithm
  - With a fixed communication budget achieve optimal convergence speed
  - This is relevant to any decentralized algorithm!

The flow control problem



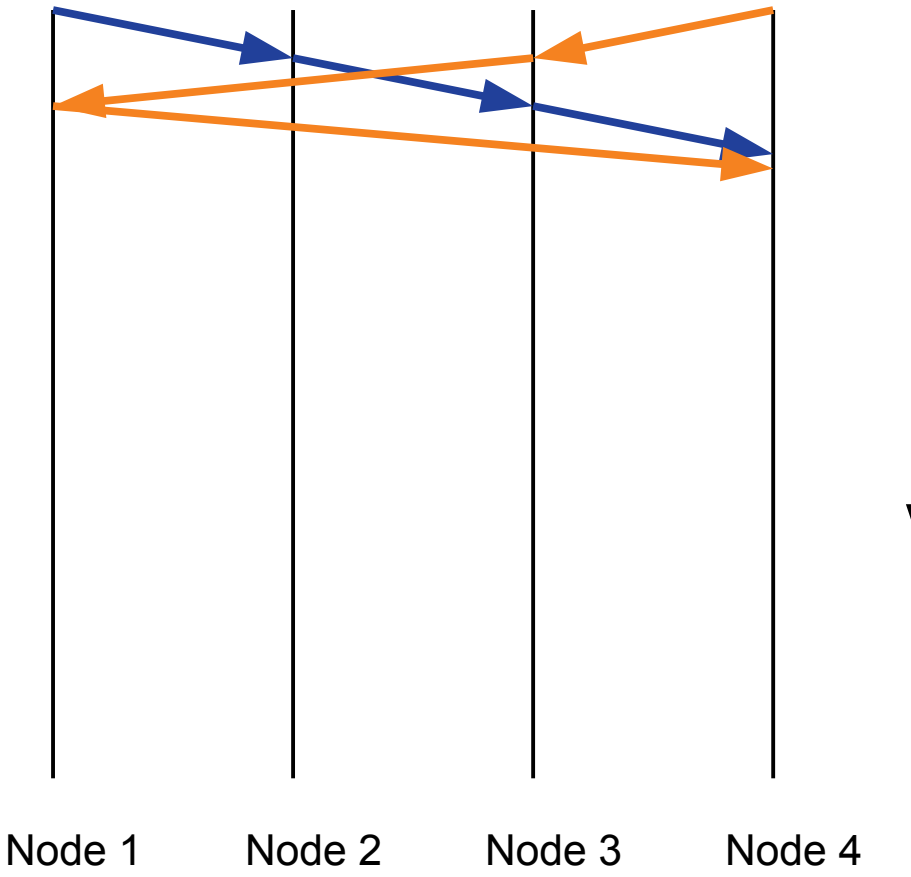
- Proactive (periodic) communication model
  - Fixed amount of traffic per period
- Good for rate limiting
  - Bursts
  - Bandwidth

The flow control problem



- Proactive (periodic) communication model
  - Fixed amount of traffic per period
- Good for rate limiting
  - Bursts
  - Bandwidth
- But slow
  - Lots of idle time

Solution: reactive?

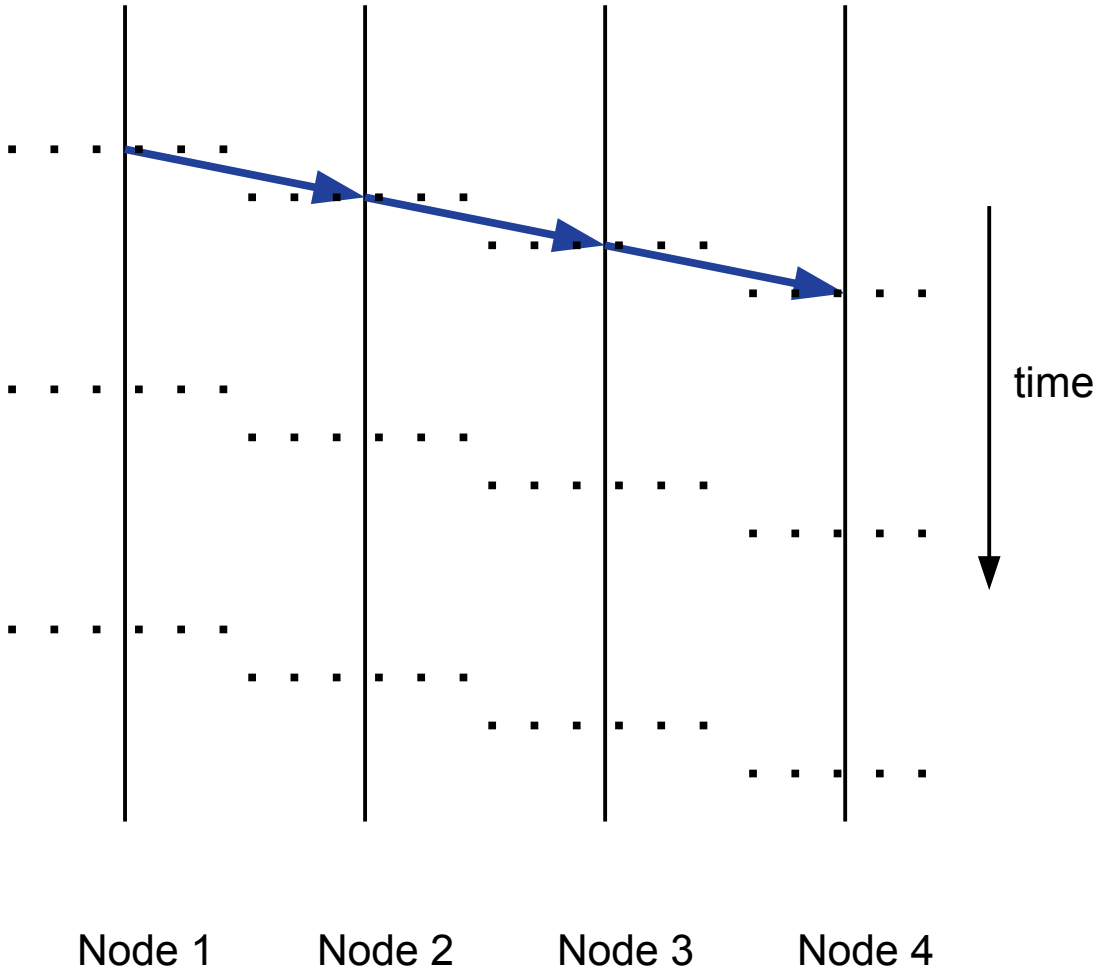


- Reactive communication model
  - React immediately
- Bad for rate limiting
  - Bursts
  - Bandwidth
- But maximal speed
- Becomes idle
  - If all messages fail

Problem statement

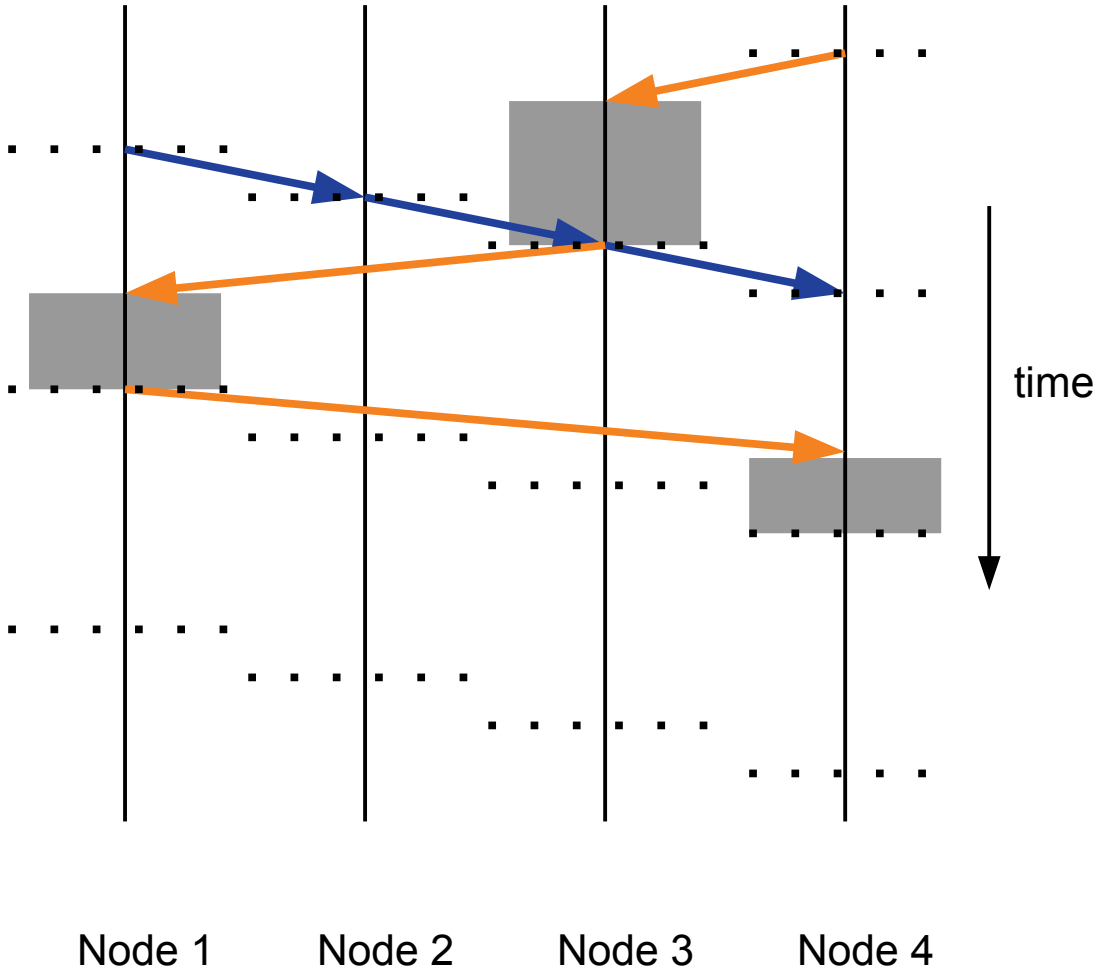
- We want a solution that has the best properties of the two approaches
  - Almost as fast as the reactive approach
  - Almost as good for rate limiting as the proactive approach
  - And avoids the system becoming idle
- We want our solution to work for as many applications as possible

Simply shifting periods?



- It is an interesting idea

Simply shifting periods?

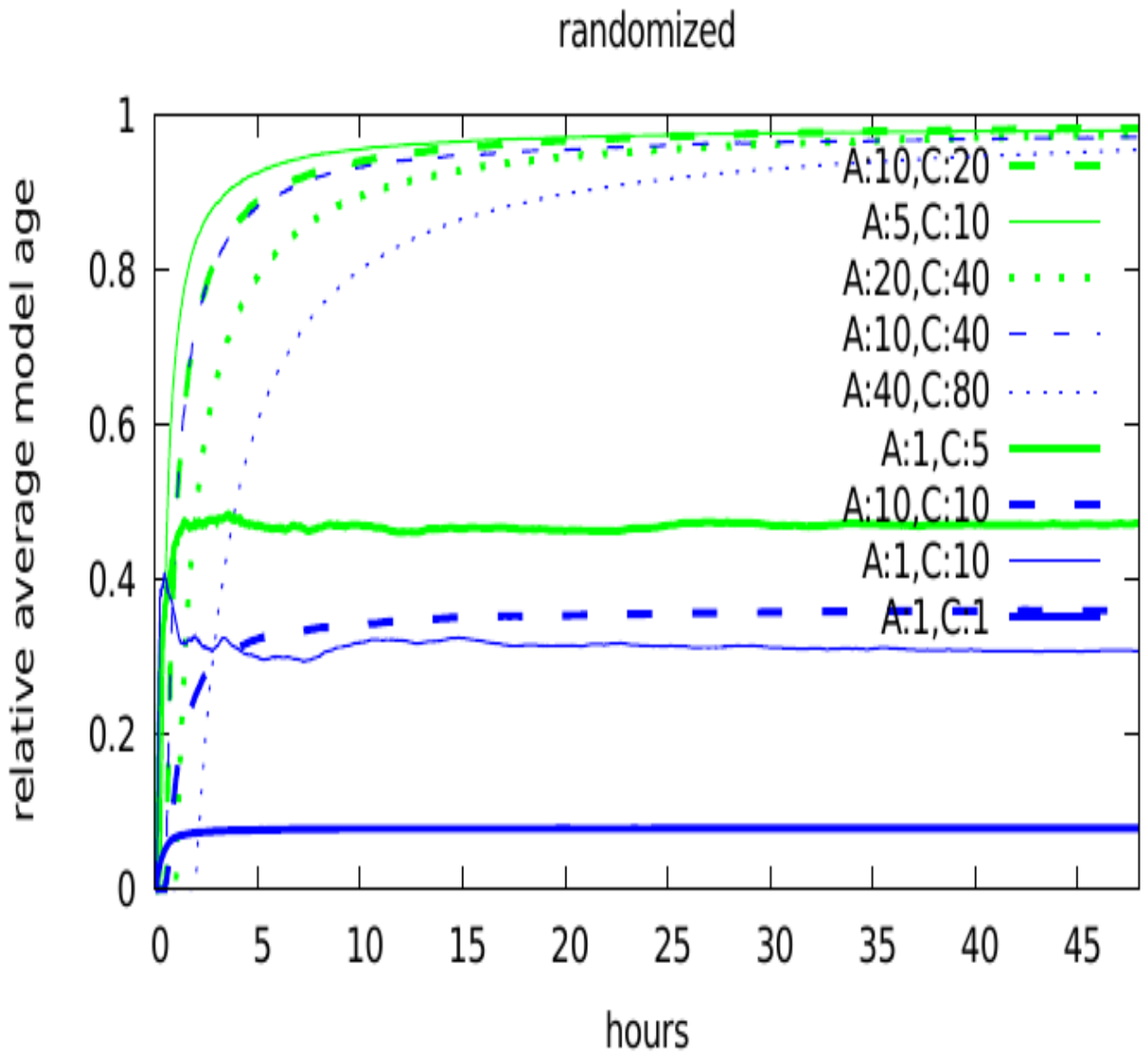


- It is an interesting idea
- But it does not work
  - Different chains of messages need different shifts
- Nevertheless it gives us a useful intuition
  - Try to allow message chains to flow while keeping rate control

Token account algorithm

- Every node receives one token in each period
- The reaction to incoming messages is immediate if there are non-zero tokens at the cost of “spending” tokens
- Similar to token bucket but
  - It is in the application layer with hooks to incorporate application semantics
  - We generalize it and offer a spectrum of algorithms between proactive and reactive
  - The proactive element offers fault tolerance too (avoiding idle state)

Gossip learning result



- We can use the network only 10% of the time
- Yet we can reach almost 100% speed for the random walks
  - Fewer models walk but faster on a close-to-optimal path
- Not obvious hot to do this for centralized federated learning!

Conclusions

- Performing most of the learning on user devices
  - Allows for combining all the local data (as opposed to isolated data silos)
  - Allows for increased privacy
  - Is very cost effective
- One can implement the entire learning process on the edge with a competitive performance
  - Optimizations like token account flow control
  - Or parameter sampling and other compression methods
- Security can also be addressed (was not discussed here)
  - Differential privacy, cryptographic techniques, etc

References

1. Gábor Danner and Márk Jelasity. Robust decentralized mean estimation with limited communication. In Marco Aldinucci, Luca Padovani, and Massimo Torquati, editors, Euro-Par 2018, volume 11014 of Lecture Notes in Computer Science, pages 447–461. Springer International Publishing, 2018. (doi:10.1007/978-3-319-96983-1_32)

2. Gábor Danner and Márk Jelasity. Token account algorithms: The best of the proactive and reactive worlds. In Proceedings of The 38th International Conference on Distributed Computing Systems (ICDCS 2018), pages 885–895. IEEE Computer Society, 2018. (doi:10.1109/ICDCS.2018.00090)

3. Gábor Danner, Árpád Berta, István Hegedűs, and Márk Jelasity. Robust fully distributed mini-batch gradient descent with privacy preservation. Security and Communication Networks, 2018:6728020, 2018. (doi:10.1155/2018/6728020)

4. Edward Tremel, Ken Birman, Robert Kleinberg, and Márk Jelasity. Anonymous, fault-tolerant distributed queries for smart devices. ACM Transactions on Cyber-Physical Systems, 3(2):16:1–16:29, October 2018. (doi:10.1145/3204411)

5. Árpád Berta and Márk Jelasity. Decentralized management of random walks over a mobile phone network. In Proceedings of the 25th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP'17), pages 100–107, St. Petersburg, Russia, 2017. IEEE Computer Society. (doi:10.1109/PDP.2017.73)

6. Árpád Berta, István Hegedűs, and Márk Jelasity. Dimension reduction methods for collaborative mobile gossip learning. In Proceedings of the 24th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP'16), pages 393–397, Heraklion, Greece, 2016. IEEE Computer Society. (doi:10.1109/PDP.2016.20)

7. István Hegedűs, Árpád Berta, Levente Kocsis, András A. Benczúr, and Márk Jelasity. Robust decentralized low-rank matrix decomposition. ACM Transactions on Intelligent Systems and Technology, 7(4):62:1–62:24, May 2016. (doi:10.1145/2854157)