THE UNIVERSITY OF CHICAGO

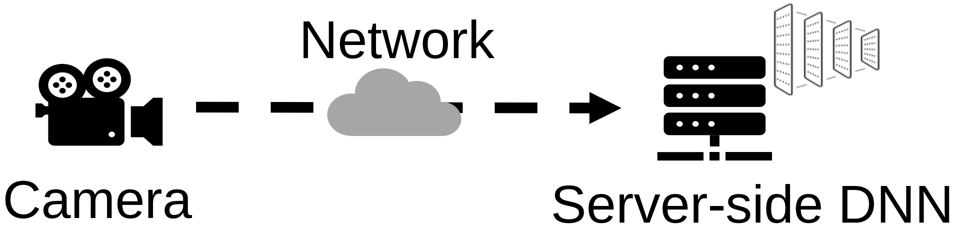
ACM Symposium on Cloud Computing

Minimizing Packet Retransmission for Real-Time Video Analytics

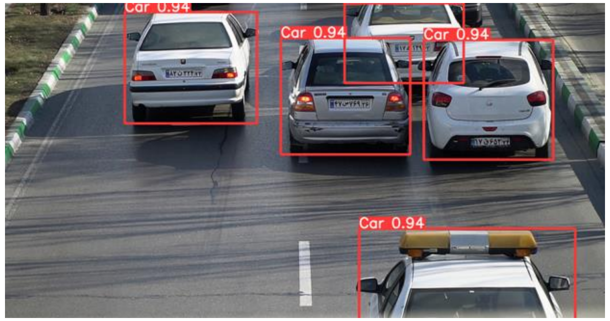
Haodong Wang, Kuntai Du, Junchen Jiang
University of Chicago

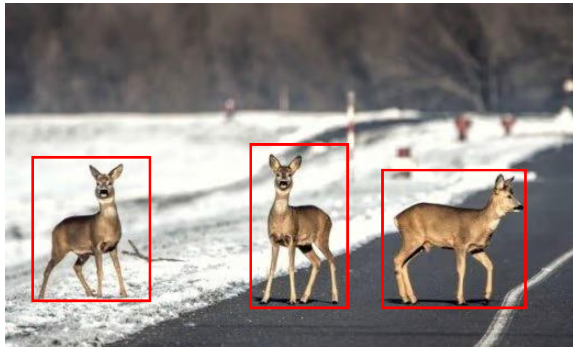
High-quality video analytics (VA)

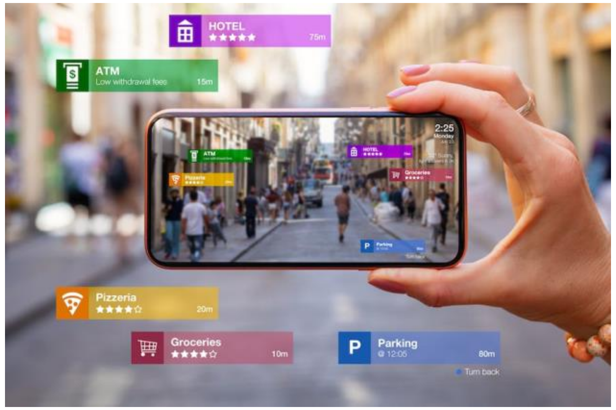
- In VA, videos collected by sensors are transmitted to cloud servers to run DNN-based inference



- It is used in many scenarios

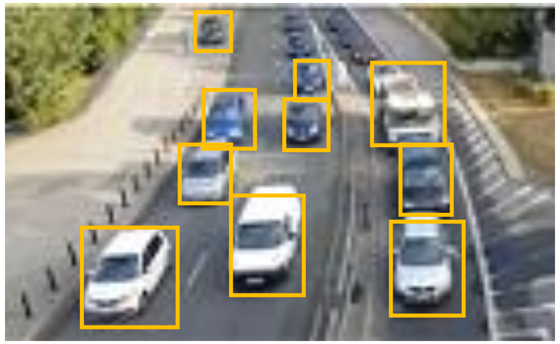




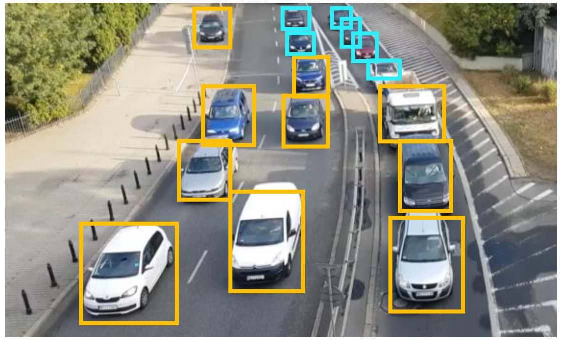


VA requires high accuracy and low delay

- High accuracy: the analysis results are close to that using uncompressed videos
  - Example: Car detection



Using a low-quality video



Using an uncompressed video

- Low delay: we can get the results in near real time

Video conferencing: <100ms

Augmented reality: <110ms

Oztas, Basak, et al. "A study on the HEVC performance over lossy networks." *2012 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012)*. IEEE, 2012.
Westphal, Cedric. "Challenges in networking to support augmented reality and virtual reality." *IEEE ICNC* (2017).

Our Idea

Application-layer designs

To reduce packet retransmission by only sending the most relevant frames to applications determined *before* transmission

Traditional Transport-layer designs

To reduce packet retransmission using additional information *irrelevant to* applications and generated *before* transmission

Our transport-layer design: T4V

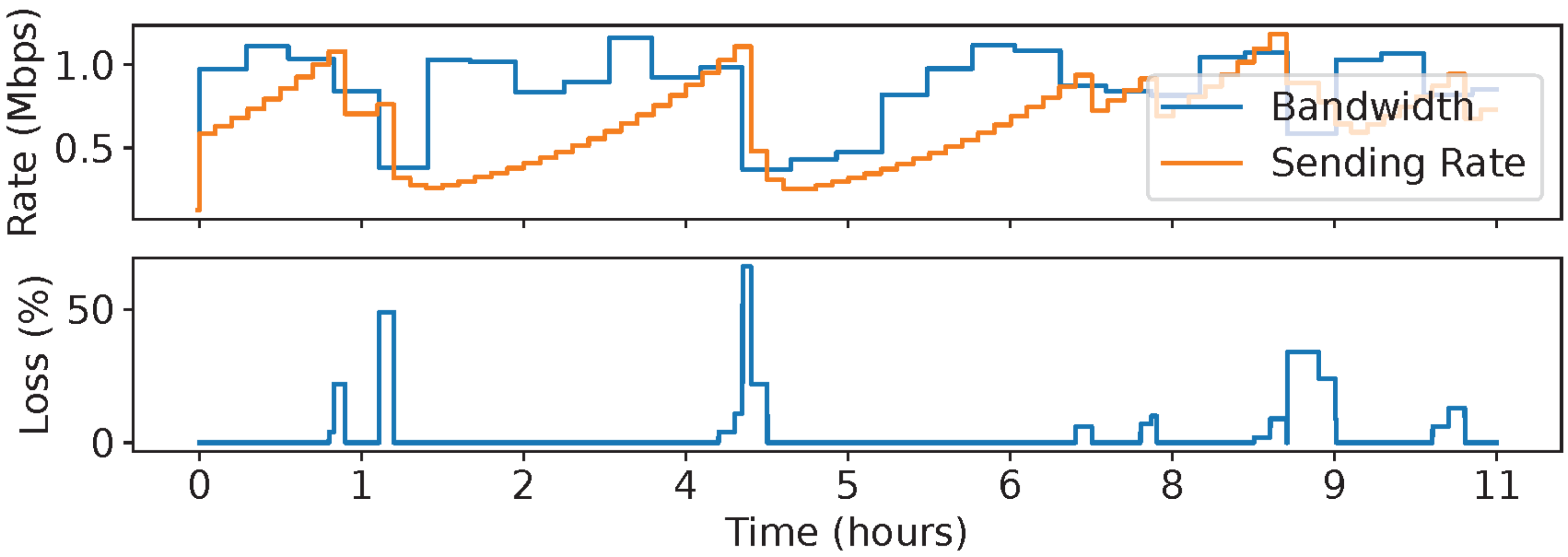
To reduce packet retransmission using *application-aware* additional information determined *during* transmission

Application-layer designs

- Idea
  - To aggressively compress video frames, and only send the most relevant frames
- Examples
  - AWStream [SIGCOMM'18]
  - DDS [SIGCOMM'20]
  - Reducto [SIGCOMM'20]
- Drawbacks
  - ***Reducing delay means sending fewer bits?***
  - ***The impact of each video frame on DNN inference can be precisely determined before transmission?***

Sending fewer bits?

- Sending fewer bits does ***not necessarily*** reduce the delay
  - Lowering bitrate can't eliminate transient ***packet losses***.
  - WebRTC Example: Bandwidth drops can cause transient packet losses because of the hysteresis of sending rate adjustment



Estimating the impact of video frames?

- The estimation of video frame impact on DNN inference ***before*** transmission is ***inaccurate***
  - Video frame impact can ***only*** be precisely obtained ***during*** transmission

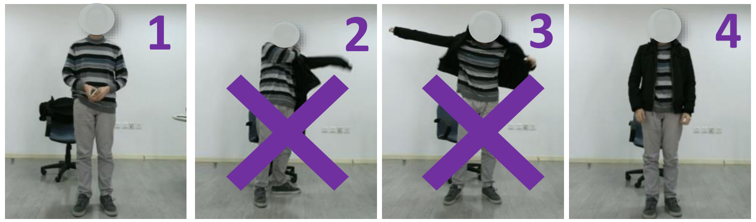
Estimating the impact of video frames? (cont.)

Example: Video frame impact can ***only*** be precisely obtained ***during*** transmission

Case 1

Frame 2, 3 lost

Inference result: standing ☒



Frame 2 has a high impact on inference results

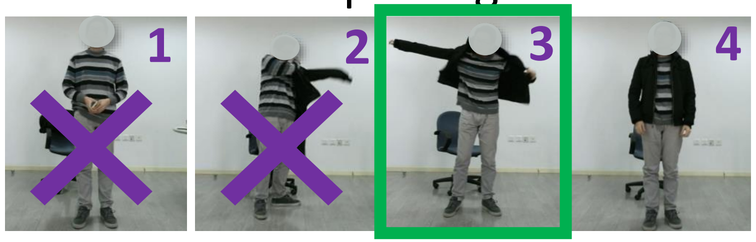
Retransmit frame 2 ☑

Inference result: putting on a coat ☑

Case 2

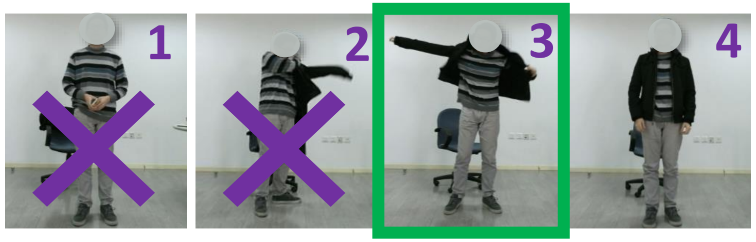
Frame 1, 2 lost

Inference result: putting on a coat ☑



Frame 2 has a low impact on inference results

Not retransmit frame 2 ☒



Inference result: putting on a coat ☑

Data source: Liu Chunhui, Hu Yueyu, Li Yanghao, Song Sijie, and Liu Jiaying. 2017. PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding. arXiv preprint arXiv:1703.07475 (2017).

Application-layer designs (revisited)

- Idea
  - To aggressively compress video frames, and only send the most relevant frames
- Examples
  - AWStream [SIGCOMM'18]
  - DDS [SIGCOMM'20]
  - Reducto [SIGCOMM'20]
- Drawbacks
  - Sending fewer bits does ***not necessarily*** reduce the delay ☒
  - The estimation of video frame impact on DNN inference ***before*** transmission is ***inaccurate*** ☒

Traditional Transport-layer designs

- Idea
  - To reduce packet retransmission by additional information
- Examples
  - Forward-error correction (FEC)
  - Bounded-loss transport
  - Selective retransmission
- Drawbacks
  - The additional information is still determined ***before*** transmission ☒
  - The additional information is ***irrelevant to*** the frame loss impact on DNN inference ☒

Our Idea (revisited)

Application-layer designs

To reduce packet retransmission by only sending the most relevant frames to applications determined *before* transmission

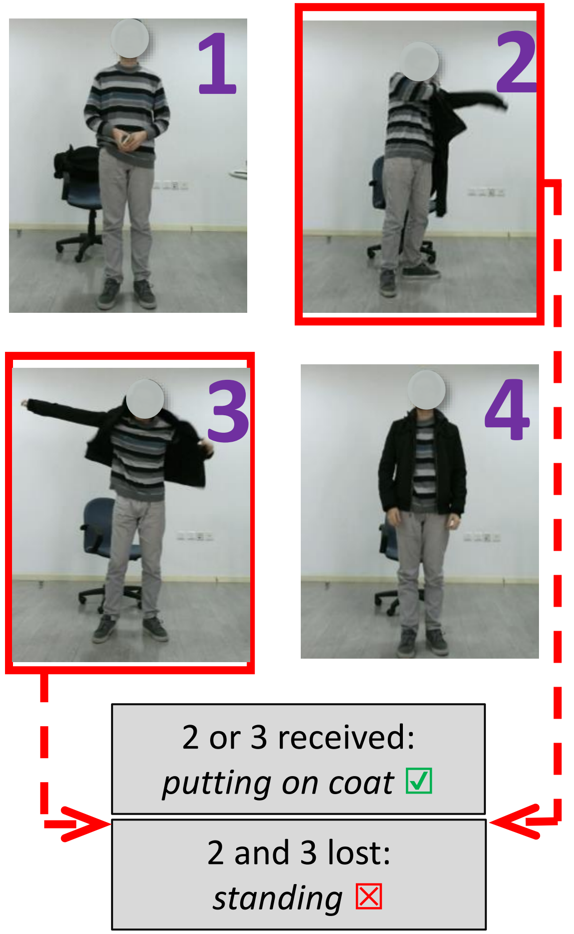
Traditional Transport-layer designs

To reduce packet retransmission using additional information *irrelevant to* applications and generated *before* transmission

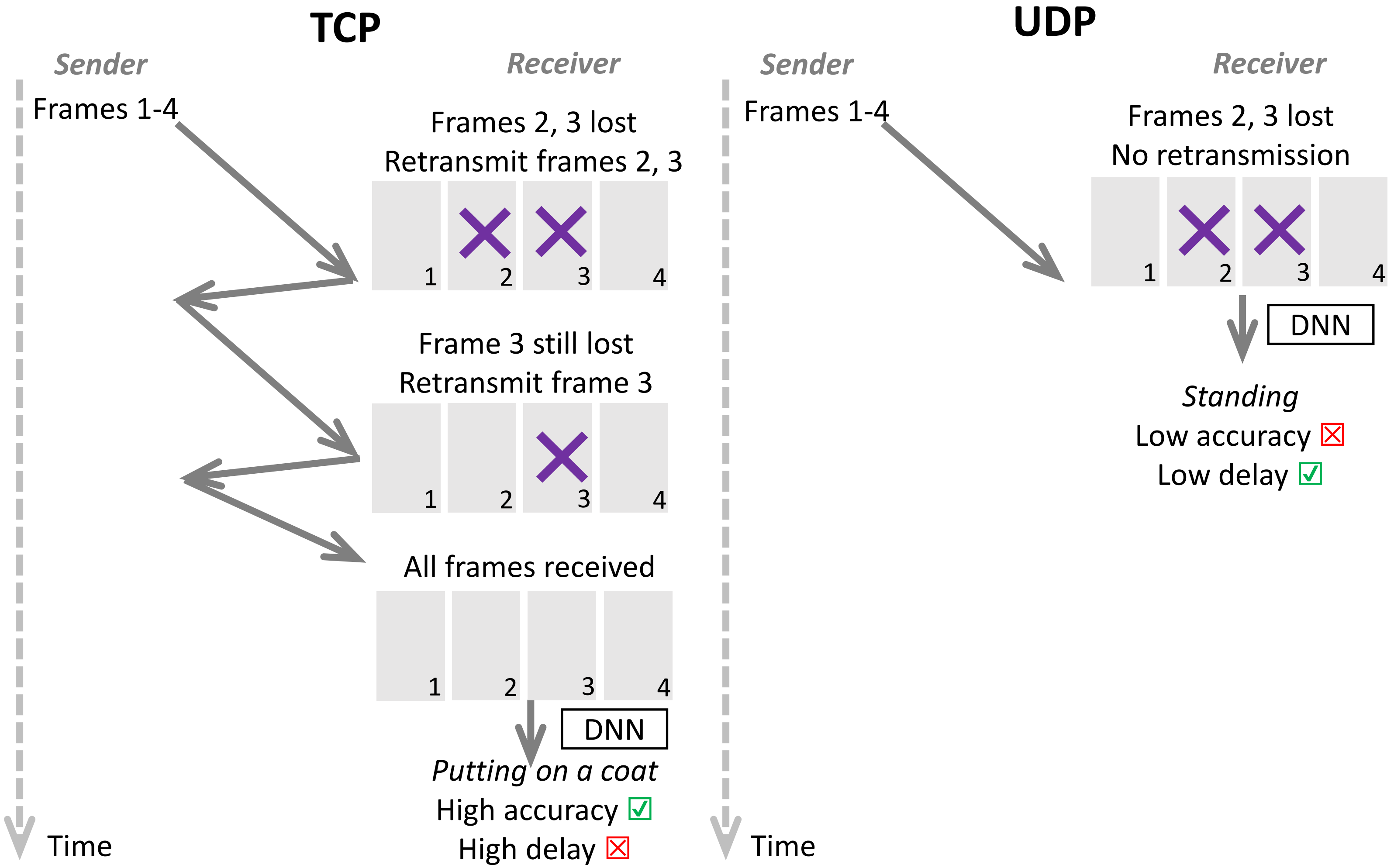
Our transport-layer design: T4V

To reduce packet retransmission using *application-aware* additional information determined *during* transmission

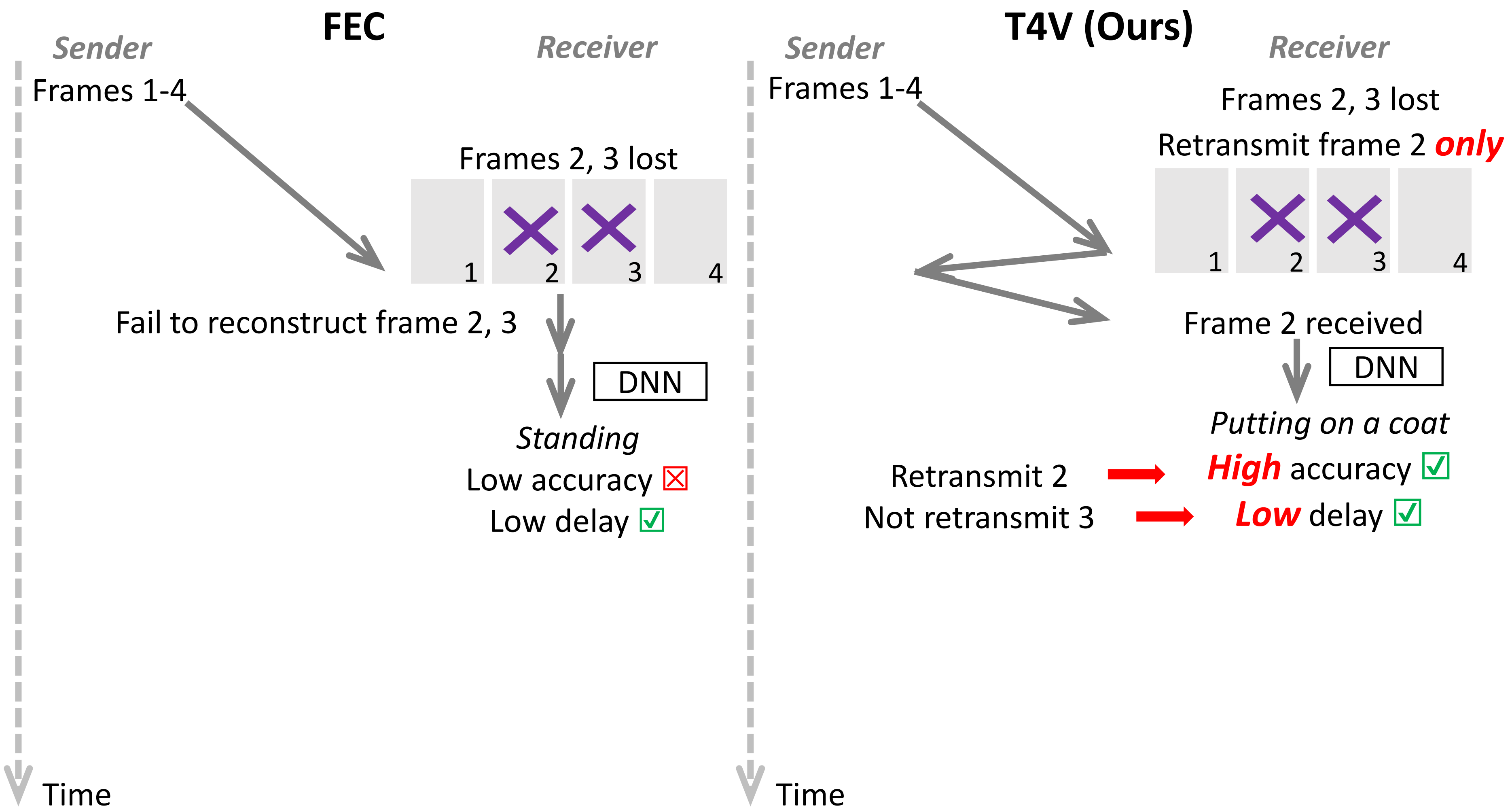
Improvement brought by our idea



Frame loss rate: 50%



Improvement brought by our idea (cont.)



Design of T4V

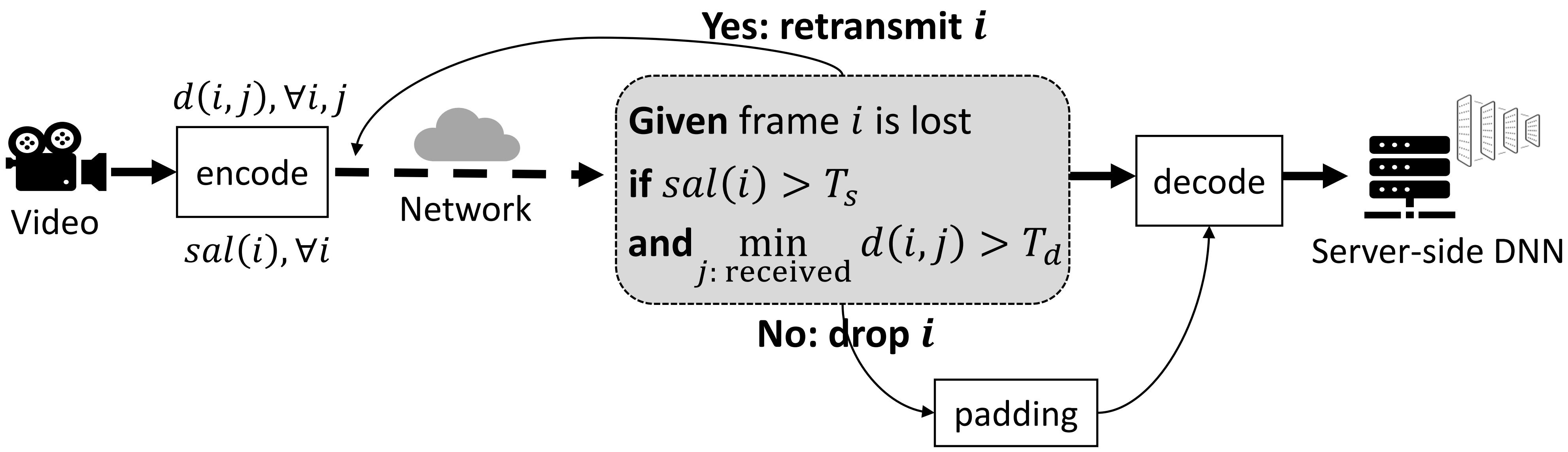
- Key Idea: ***Incremental impact*** of each frame ***conditioned on*** received frames

  - Definition: *Given the received frames*, how much obtaining or losing a frame would change the inference result

  - Components
    - Frame difference: Pixel-wise difference between frames
    - Saliency: Pixel-wise accumulation of the gradient of the inference result with respect to the frame [Open Question 1]

Design of T4V (cont.)

- How to use the incremental impact
  - $d(i, j)$: *frame difference* between frame i and j
  - $sal(i)$: *saliency* value of frame i
  - T_s, T_d: user-defined *thresholds*



[Open Question 2]

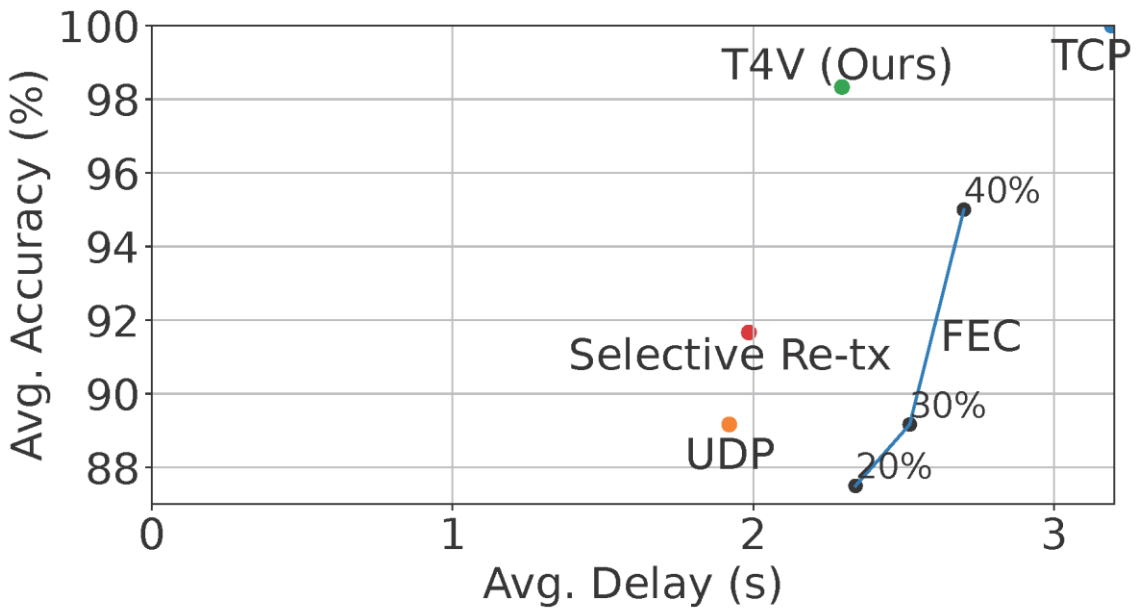
Open Questions

- Saliency estimation
  - The ***overhead*** to get ***accurate*** saliency values is high (forward propagation and backward propagation on a large DNN)
  - Direction: saliency values can be reasonably ***approximated*** by training ***cheap predictors***

- Faster retransmission decisions
  - Retransmission ***decisions*** require nontrivial ***computation***
  - Direction 1: to ***offload*** some compute to sensors
  - Direction 2: to ***pipeline*** packet retransmission with DNN inference on the received frames

Case Study

- VA application: action recognition
- DNN: I3D
- Network simulation
  - Streaming delay = size of transmitted packets / bandwidth (200Kbps)
  - Each frame is sent in one packet (1.5KB, consistent with the average frame size in a low-quality video (e.g., 360p))
  - Frame loss rate: 30%
  - T_s = 0.01 and T_d = 0.001
  - 100 rounds of independent tests
- Data: 12 video clips from Kinetics-400
  - 32 frames per video clip
- Baselines: TCP, UDP, FEC, and H.264-based selective retransmission



- T4V vs. TCP: Similar accuracy with 30%+ less packet retransmissions
- T4V vs. UDP: inaccuracy reduced from 11% to 2% at marginal delay inflation (15%)
- T4V vs. selective retransmission: reduces inaccuracy from 8% to 2% with only 10% delay increase

Conclusions

- We propose a transport-layer design, T4V, for real-time video analytics.
- T4V makes a case for deciding whether to retransmit a frame based on its incremental impact on inference output conditioned on received frames.
- Our contribution is a framework to make retransmission decisions based on the incremental impact per frame, and a case-study evaluation to quantify its potential benefit.