

AI-Enhanced Transport Layer and End-to-End Intelligence

Lecture 7

By Dr. Tarfa Yaseen Hamed

Apr. 2026

Outline

- ▶ Introduction
- ▶ The Pillar: AI-Driven Congestion Control (Deep-TCP)
- ▶ Case Study 1: Distinguishing Congestion from Noise
- ▶ Case Study 2: Intelligent Retransmission (Deep-ARQ at Layer 4)
- ▶ Performance Comparison: AI vs. Traditional Transport
- ▶ Technical Synthesis: Cross-Layer Transport (The "Intelligence Feed")

Introduction: The End-to-End Responsibility

The Transport Layer (Layer 4) ensures that data is delivered error-free, in sequence, and without loss between the source and destination applications.

- **The Classic Duo:** TCP (Reliable, Connection-oriented) and UDP (Fast, Connectionless).
- **The 2026 Challenge:** Traditional TCP was designed for wired networks where packet loss meant "**congestion**." In 6G wireless/satellite networks, packet loss often means "**interference**" or "**high mobility**."
- **The Problem:** Traditional TCP reacts to wireless noise by slowing down speed unnecessarily, causing a massive drop in throughput.

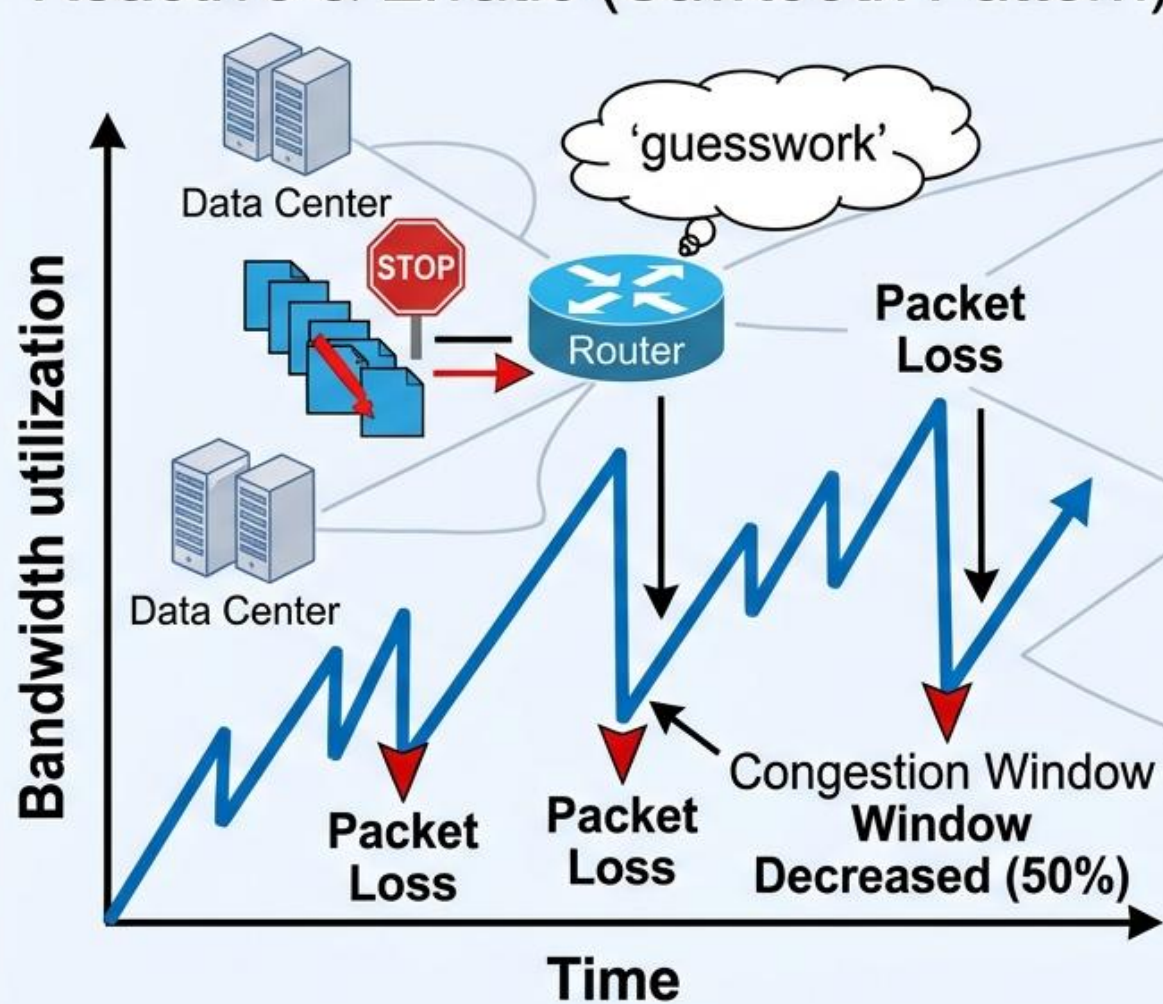
The Pillar: AI-Driven Congestion Control (Deep-TCP)

Traditional TCP uses mathematical formulas to increase/decrease the "Congestion Window" CWND.

Congestion Window: It is a variable used by the sender to determine how many packets it can send into the network at once without waiting for an acknowledgment (ACK).

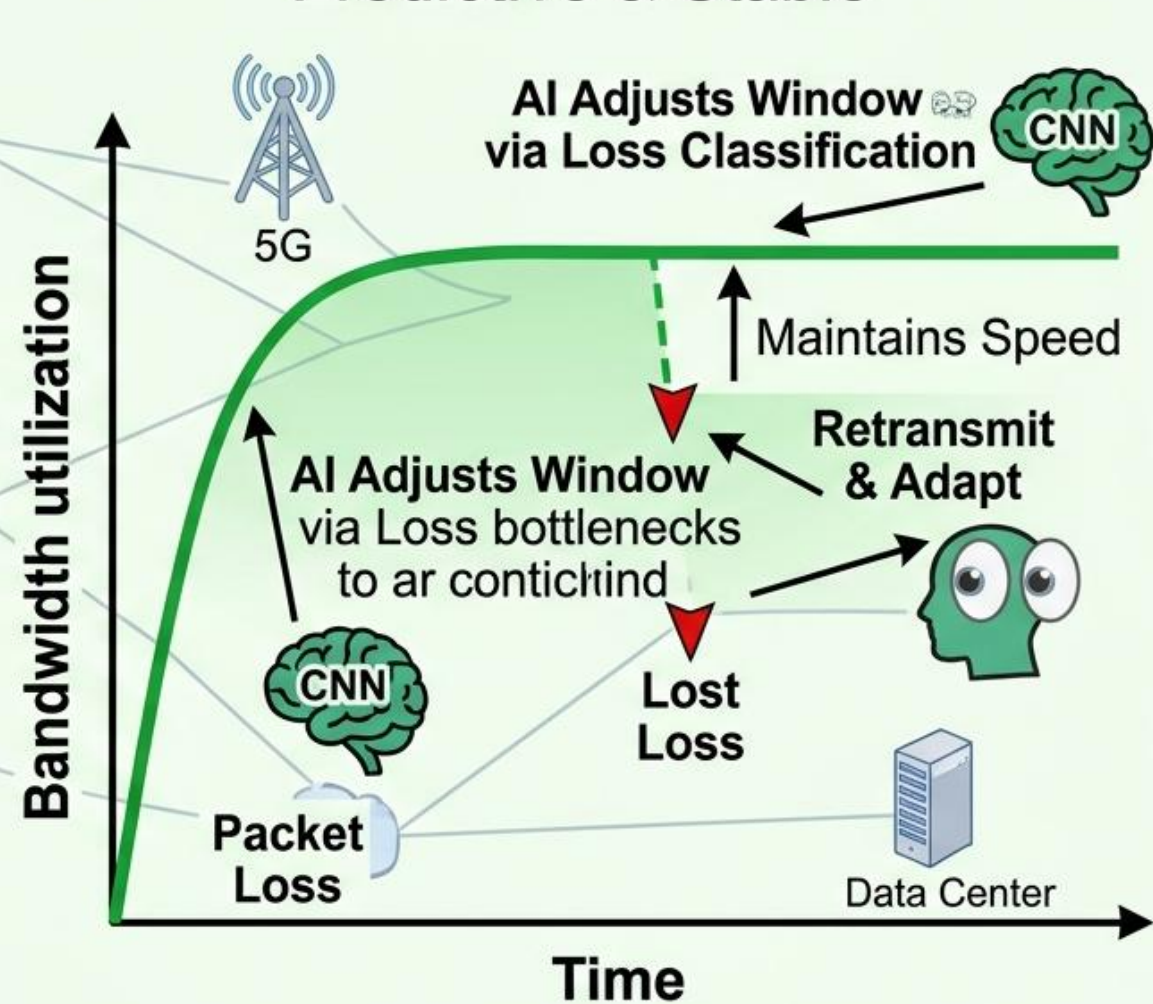
- **The AI Approach:** Use Deep Reinforcement Learning (DRL) to manage the window size.
- **The Agent:** The Transport Protocol stack on the sender's device.
- **The State (s):** RTT (Round Trip Time) variations, packet arrival rates, and inter-arrival jitter.
- **The Action (a):** Precisely adjusting the CWND and the "Pacing Rate" (how fast packets are injected into the network).
- **The Reward (r):** Maximizing Kleinrock's Power Metric ($Throughput_4 / Delay$).

Traditional TCP (Cubic): Reactive & Erratic (Sawtooth Pattern)



Traditional TCP (Cubic):
Reactive & Erratic (Sawtooth Pattern)

AI-Native Transport (Deep-TCP): Predictive & Stable



AI-Native Transport (Deep-TCP):
Predictive & Stable

Traditional Method (The "Guessing" Game):

Traditional TCP uses a "Trial and Error" approach:

1. **Slow Start:** It starts with a small window (e.g., 10 packets).
2. **Increase:** If the packets arrive safely, it doubles the window. It keeps growing until...
3. **The Crash:** A packet is lost. Traditional TCP assumes the network is "clogged" and immediately cuts the Congestion Window by 50%.
 - **Result:** This creates the Sawtooth Pattern we saw in Figure above, where the speed is constantly crashing and restarting.

AI-Driven Method (The "Knowing" Game):

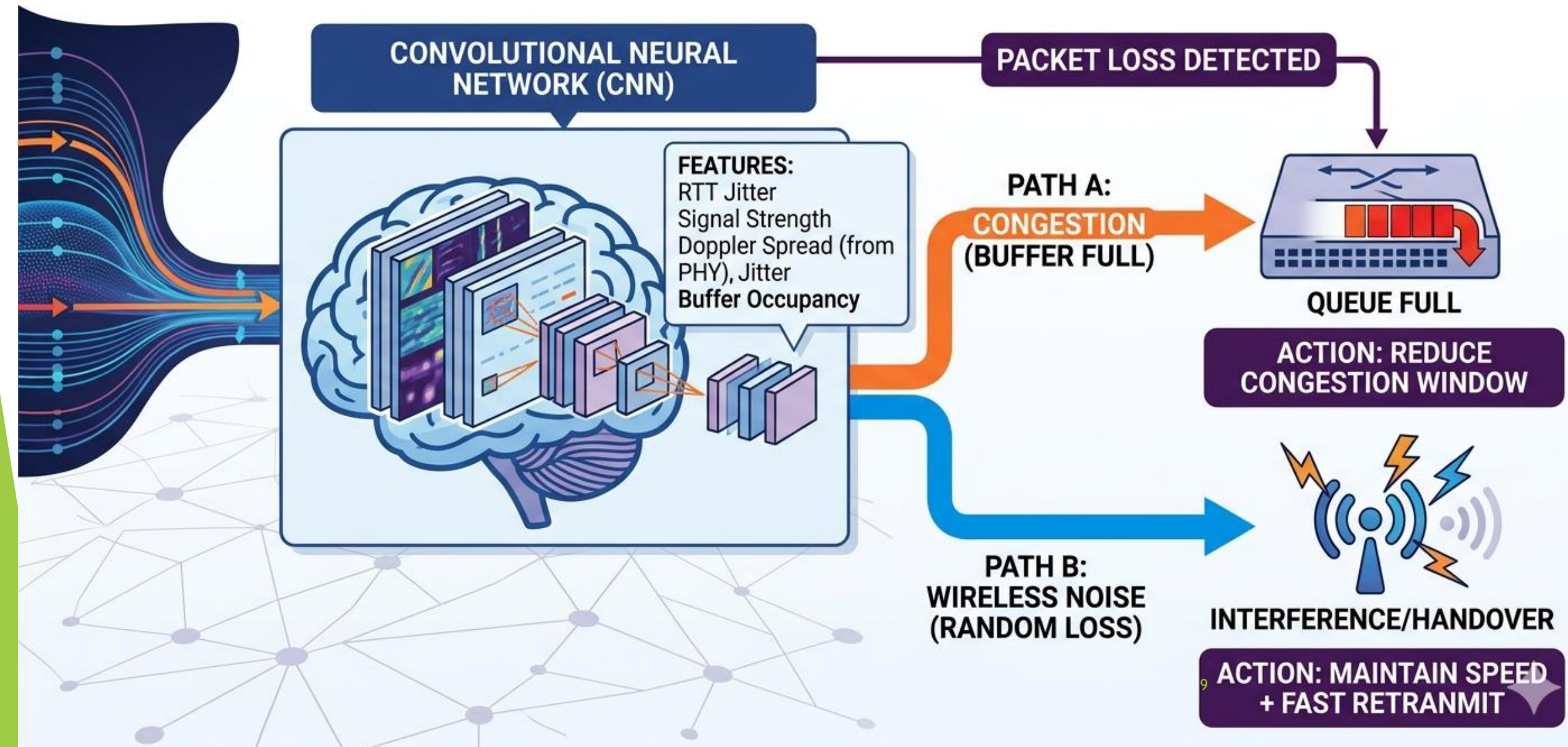
Deep-TCP uses Reinforcement Learning to manage the CWND:

- Instead of waiting for a crash, the AI looks at **RTT (Round Trip Time) Jitter** and **CSI (Channel State Information)**.
- If it sees the network getting "jittery," it slightly shrinks the window *before* a packet is lost.
- **Result:** A smooth, stable throughput that stays at the maximum capacity of the 6G link.

Case Study 1: Distinguishing Congestion from Noise

- **The Scenario:** A user on a High-Speed Train (300 km/h) experiencing frequent handovers and interference.
- **Traditional Failure:** TCP sees a packet drop (due to a satellite handover) and assumes the network is congested. It cuts the speed by 50%.
- **The AI Solution (CNN Classifier):** A Convolutional Neural Network analyzes the "shape" of the delay patterns.
- **The Outcome:** The AI identifies that the loss was "Random/Wireless" and NOT "Congestion." It tells TCP to *maintain* speed and just retransmit the lost packet, keeping the 5G/6G link at 100% efficiency.

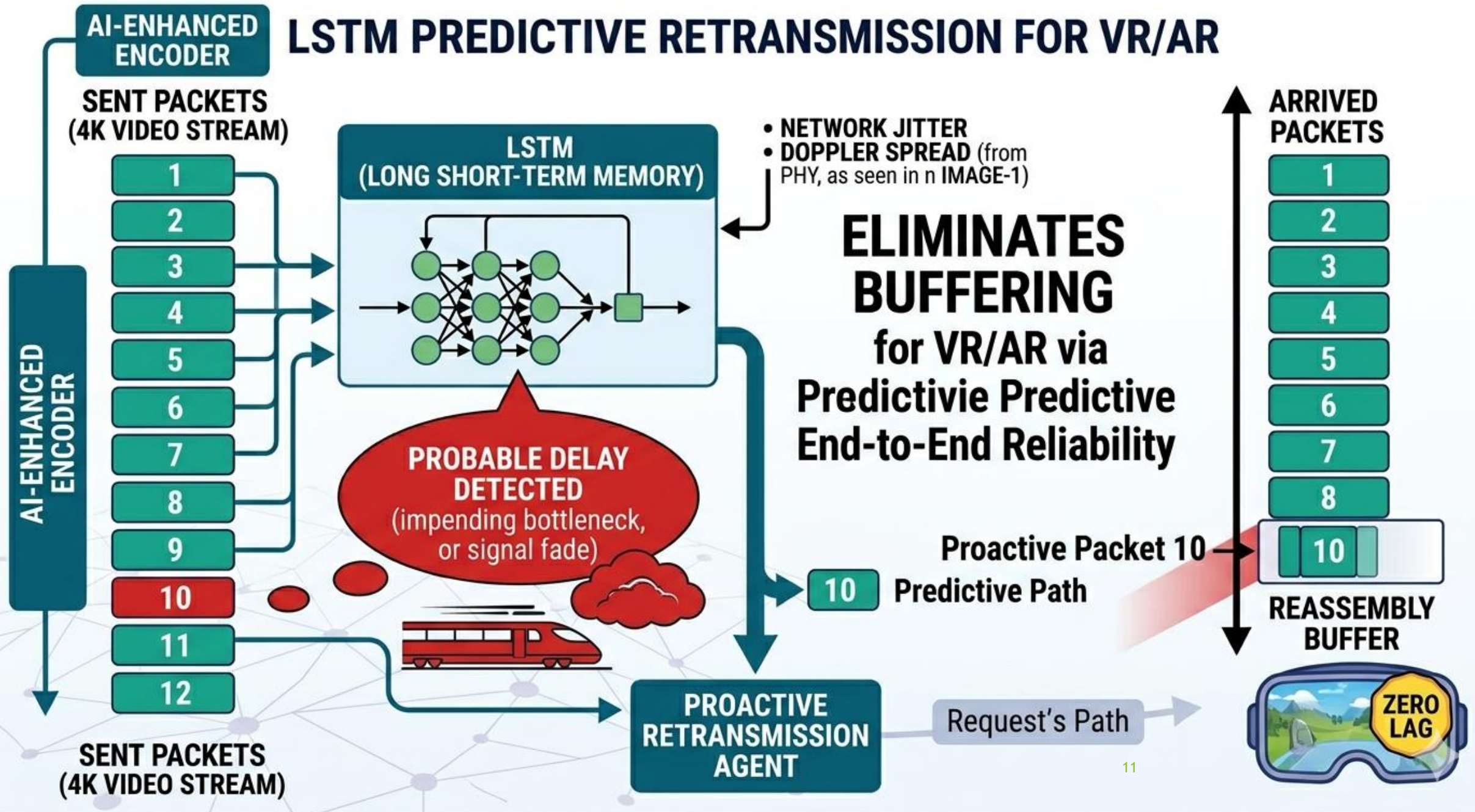
CNN LOSS CLASSIFIER: DISTINGUISHING CONGESTION FROM NOISE



Case Study 2: Intelligent Retransmission (Deep-ARQ at Layer 4)

- While it was observed the ARQ at the Data Link Layer, Layer 4 handles **End-to-End ARQ**.
- Traditional **ARQ (Automatic Repeat Request)** is a simple "Feedback Loop": the receiver gets a corrupted packet, sends a "Negative Acknowledgment" (NACK), and the sender retransmits.
- **Deep-ARQ** replaces this reactive logic with a **Deep Learning** model (LSTM) that predicts errors before they occur or makes "intelligent" decisions about when and how to retransmit.
- **The Pillar: LSTM-based Predictive Retransmission.**
- **The Solution:** An LSTM predicts which segments of a 4K video stream are likely to arrive late based on the current "Network Jitter."
- **The Outcome:** The sender sends a "Proactive Duplicate" of the critical data before the receiver even asks for it.
- **The Result:** Eliminates "buffering" circles in video streaming and VR.

LSTM PREDICTIVE RETRANSMISSION FOR VR/AR



Performance Comparison: AI vs. Traditional Transport

Metric	Traditional TCP (Cubic)	AI-Native Transport (Deep-TCP)
Reaction to Loss	Always slows down (Reactive)	Analyzes cause before reacting (Predictive).
Throughput	Fluctuating (Sawtooth pattern)	Smooth and maximized (Stable).
Startup Speed	Slow "Slow-Start" phase	Instant "Optimal-Start" via history.
SLA Compliance	Low (High Jitter)	High (Deterministic Latency).

Technical Synthesis: Cross-Layer Transport (The "Intelligence Feed")

In 2026, Layer 4 is no longer "blind" to the layers below:

- **Input from Network Layer:** The router informs the Transport Layer about a "Path Change" via **ECN (Explicit Congestion Notification)** enhanced by AI.
- **Input from DLL:** The MAC layer reports **Doppler Spread** to the Transport Layer.
- **Decision:** The Transport Layer switches from TCP to a "Reliable-UDP" variant automatically when it senses high-mobility conditions.

3D CROSS-LAYER 'INTELLIGENCE FEED' FOR AI-DRIVEN TRANSPORT LAYER

