

[Click here](#) for X.25 Wikipedia article

X.25 packets vs Ethernet packets

X.25 packets and Ethernet packets operate at different layers of the network architecture, with X.25 functioning as a connection-oriented network/packet layer protocol for wide area networks and Ethernet functioning as a connectionless data link/physical layer protocol typically for local area networks.

Core Differences

- **[OSI Layer:](#)**
 - **X.25 Packet:** Operates at Layer 3 (the Packet Layer / Network Layer), managing end-to-end virtual circuits, flow control, and multiplexing.
 - **Ethernet Frame/Package:** Operates at Layers 1 and 2 (Physical and Data Link Layers), handling node-to-node physical delivery and MAC addressing over a local network medium.
- **Connection Type:**
 - **X.25:** Connection-oriented. It establishes a virtual call or circuit before data moves, meaning sequential packets follow a predefined path.
 - **Ethernet:** Connectionless. Each frame contains full source and destination MAC addresses and is broadcast or switched independently without a pre-established circuit.
- **Primary Environment:**
 - **X.25:** Designed for noisy, error-prone Wide Area Networks (WANs) running over analog or low-speed digital telecommunication lines.
 - **Ethernet:** Built for high-speed Local Area Networks (LANs) using copper or fiber cables.
- **Size and Overhead:**
 - **X.25 Packets:** Typically small (128 or 256 bytes) to minimize latency on error-checked, low-speed links.
 - **Ethernet Frames:** Considerably larger (standard payloads up to 1,500 bytes or more via jumbo frames).

Comparison Table

Feature	X.25 Packet	Ethernet Frame / Packet
OSI Layer	Layer 3 (Network / Packet)	Layer 1 & 2 (Physical & Data Link)
Scope	Wide Area Network (WAN)	Local Area Network (LAN)
Connection State	Connection-oriented (Virtual Circuits)	Connectionless (Datagram/Frame delivery)

Feature	X.25 Packet	Ethernet Frame / Packet
Error & Flow Control	Built heavily into the protocol stack (LAPB at Layer 2)	Minimal at Layer 2 (relies on higher layers like TCP for robust recovery)
Status	Legacy technology, mostly replaced by IP/MPLS	Modern and dominant standard for wired networking

How does x25 handle flow control and multiplexing

X.25 handles flow control and multiplexing natively at **Layer 3 (the Packet Layer)**, which allowed it to reliably manage multiple data streams over the slow, error-prone analog lines of early wide-area networks.

Here is how both mechanisms function:

1. Multiplexing via Logical Channels

X.25 achieves multiplexing by dividing a single physical communication link into thousands of independent, simultaneous bidirectional data streams called **Logical Channels**.

- **Logical Channel Identifiers (LCIs):** Every X.25 packet header contains a **Logical Channel Group Number (LCGN)** and a **Logical Channel Number (LCN)**. Together, these form a unique 12-bit identifier allowing up to **4,096 distinct logical channels** on a single physical line.
- **Virtual Circuits:** These channels are used to establish either Permanent Virtual Circuits (PVCs, which act like dedicated leased lines) or Switched Virtual Circuits (SVCs, which are spun up and torn down like phone calls).
- **The Process:** When a device sends data, the X.25 packet layer tags the packet with the assigned LCN. The network switches read this LCN to route the packet along the pre-established path, ensuring that traffic from entirely different users or applications can share the same physical cable without interfering with one another.

2. Flow Control via Sliding Window Protocol

Because early network lines were slow, X.25 used a rigid, hop-by-hop **sliding window protocol** at the packet layer to ensure a fast sender could not overwhelm a slow receiver or congest the network.

- **Sequence Numbers:** Each data packet is assigned a send sequence number, **P(S)**, and an expected receive sequence number, **P(R)**. By default, X.25 uses standard 3-bit sequence numbers (modulo 8, allowing window sizes up to 7), but it can be extended to 7-bit numbers (modulo 128) for long-delay satellite links.
- **Window Size (W):** The window size defines how many packets a device can send ahead without receiving an acknowledgment. If the window size is 3, the sender can transmit 3 packets and must then pause until the receiver acknowledges them.
- **Flow Control Packets:** Control is maintained using specific packet types:
 - **Receive Ready (RR):** Sent by the receiver to acknowledge received packets and signal that it is ready to accept the next window of data.

- **Receive Not Ready (RNR):** Sent by the receiver to temporarily halt transmission (e.g., if its internal buffers are full). Once it clears its queue, it sends an RR packet to resume data flow.

Compare [X.25](#) (published in 1976) with [ATM](#) (published in 1990)

X.25 and **Asynchronous Transfer Mode (ATM)** are both connection-oriented, packet-switching technologies designed for wide-area networking, but they represent two entirely different eras of hardware capability.

While **X.25** was built in the 1970s for slow, unreliable copper lines that required heavy error-checking, **ATM** was engineered in the late 1980s and 1990s for blazing-fast, highly reliable fibre-optic networks.

Core Differences

- **OSI Layer Operating Model:**
 - **X.25:** A full Layer 3 (Network/Packet Layer) protocol stack that manages heavy error handling and flow control at every single hop along the network path.
 - **ATM:** Functions primarily at Layer 2 (Data Link Layer). It strips away hop-by-hop error checking and flow control, pushing those responsibilities to the end devices (Layer 4 TCP) to maximize switching speeds.
- **Packet vs. Cell Switching:**
 - **X.25:** Uses variable-length packets (typically 128 or 256 bytes). Variable sizes cause unpredictable delays at switches, which is problematic for real-time traffic.
 - **ATM:** Uses fixed-size, 53-byte packets called **cells** (5 bytes of header, 48 bytes of payload). Fixed-size cells allow switching to be done entirely in hardware (ASICs), ensuring predictable, ultra-low latency.
- **Quality of Service (QoS):**
 - **X.25:** Does not inherently support distinct QoS classes. Traffic is handled on a best-effort basis, making it poor for voice or video.
 - **ATM:** Features highly robust, native QoS categories (like Constant Bit Rate for voice, and Variable Bit Rate for video), making it the first major attempt at a true multi-service network.

How They Handle Multiplexing and Flow Control

The structural differences between the two protocols are clearly highlighted by how they handle the two core functions you asked about previously:

Multiplexing

- **X.25:** Uses a 12-bit **Logical Channel Identifier (LCI)** to multiplex up to 4,096 logical channels over a single link. Routing decisions are made at Layer 3 using packet headers.
- **ATM:** Uses a two-tiered hierarchical system in the cell header: a **Virtual Path Identifier (VPI)** and a **Virtual Channel Identifier (VCI)**. This allows millions of connections to be

bundled together into "virtual paths," allowing core network switches to route massive bundles of traffic rapidly without inspecting individual user channels.

Flow Control

- **X.25:** Enforces strict, hop-by-hop flow control using **sliding windows** and explicit control packets (RR/RNR) at every intermediate switch.
- **ATM:** Has **no hop-by-hop flow control** for its standard data traffic. Instead, it uses traffic policing at the edge of the network (e.g., discarding cells that exceed agreed bandwidth limits) and congestion notification bits (like Explicit Forward Congestion Indication, or EFCI) to warn end-devices to slow down.

Comparison Table

Feature	X.25	ATM
Primary Era	1970s–1980s	1990s–2000s
Transmission Media	Analog / Low-quality copper lines	High-speed fiber-optic lines
Data Unit Type	Variable-length packets (up to 4096 bytes)	Fixed-size 53-byte cells
Multiplexing Labels	LCGN + LCN (12-bit total)	VPI + VCI (24 to 28-bit total)
Error Handling	Hop-by-hop correction (high overhead)	Header error check only; dropped cells left to higher layers
Switching Method	Software-based packet processing	Hardware-based cell switching (ASICs)
Speed Capabilities	Low speed (typically 64 Kbps to 2 Mbps)	High speed (155 Mbps to 10 Gbps+)

Would you like to examine **how the 53-byte cell size of ATM was structurally decided** (the compromise between voice and data engineers), or are you focusing on **how modern Carrier Ethernet and MPLS networks evolved** from these systems?

What was the typical size of a single hop, say, in Australia, with X.25

In Australia's historical X.25 infrastructure—most notably Telecom Australia's public packet-switched network, [AUSTPAC](#), introduced in **1982**—the physical distance or "size" of a single hop varied drastically depending on whether the traffic was local or interstate.

Because X.25 networks are structured on a hop-by-hop basis, a "hop" refers to the physical transmission link and processing between two nodes (such as from a user terminal to a local exchange, or between core network switches).

1. The Physical Geographic "Size" of a Hop

- **Intra-city (Local) Hops:** The majority of single hops were very short, typically **under 10 to 50 kilometres**. Users dialled into a local Packet Assembler/Disassembler (PAD) over standard analog telephone lines (PSTN), meaning the first hop was just the distance from their office or home to the nearest Telecom Australia exchange.
- **Interstate (Core) Hops:** Because of Australia's massive, sparse geography, a single core-network hop between major capital cities could stretch across vast distances. For instance, a direct link from Sydney to Melbourne represented a single network hop spanning roughly **700 to 900 kilometres** of physical telecom line.

2. The Functional "Size" (Speed and Payload Constraints)

In X.25 architecture, the physical distance of a hop directly collided with severe bandwidth and data-size constraints, making long-distance hops notoriously slow.

- **Typical Link Speeds:** Local user access hops were highly constrained, usually running over modems at **300 bps, 1200 bps, or 2400 bps**. Internal core network hops between telecom switches typically operated at **4.8 Kbps, 9.6 Kbps, or up to 64 Kbps**.
- **Packet Size Overhead:** Because line speeds were so low, a single hop could not efficiently handle large amounts of data without causing severe queuing delays. Therefore, the maximum packet payload size for a single hop was strictly limited to **128 or 256 bytes**.

The Cost of a Long-Distance Hop: Latency

Because X.25 mandates strict **hop-by-hop error checking and flow control**, intermediate switches could not forward a packet until the entire packet was received and its integrity verified.

If a 256-byte packet had to make a single long-distance hop over a 9.6 Kbps link across the Nullarbor Plain (e.g., Adelaide to Perth), the serialization delay alone was massive. Combined with the processing overhead, it was common for an end-to-end journey across just a few hops in Australia to suffer from **latencies exceeding 500 milliseconds (half a second)**.

Are you researching AUSTPAC for **historical computing archives**, or are you calculating **propagation and transmission delays** for a legacy network simulation?

No, just documenting, for my info, thanks.

You're very welcome! Documenting these early network architectures is a fascinating project—it really highlights just how much engineering went into overcoming the physical and technical limits of that era.

If you ever need to dive back into early Australian telecom history, AUSTPAC technicalities, or compare other vintage networking standards, feel free to reach out.

**** End of Page**