X.25

The initial ITU standard for the X.25 packet-switching protocol was approved and published in March 1976 by the CCITT (now the International Telecommunication Union, or ITU-T). The normal packet size was 128 bytes, with each packet wrapped in a `01111110` byte, with “bit-stuffing” employed whenever the packet needed to include that bit sequence. Read more about this issue here.

Transition 5-bit Baudot Teletype in the 1930s, to 7-bit ASCII X.25 in the 1980s

The transition from **5-bit Baudot Teletype systems in the 1930s to 7-bit ASCII over X.25 networks in the 1980s** represents the foundational evolution of modern data communications. It marks the shift from electromechanical, point-to-point telegraphy to electronic, packet-switched digital networking.

Here is a direct comparison of how the core technologies evolved over those fifty years.

Architectural Evolution

Feature	1930s Baudot Teletype	1980s ASCII over X.25
Character Encoding	**5-bit Baudot-Murray code** (32 distinct values).	**7-bit ASCII** (128 distinct values) plus an optional 8th parity bit.
Character Set	Uppercase letters, numbers, and basic punctuation only; relies on "FIGS" and "LTRS" shift keys to reuse codes.	Uppercase and lowercase letters, extended punctuation, and 32 dedicated control characters.
Network Architecture	**Point-to-point** or circuit-switched telegraph networks (Telex).	**Packet-switched** wide-area networks (WAN).
Transmission Mode	Asynchronous serial, current loop (physical opening/closing of a circuit).	Layered protocol stack (ITU-T X.25) over synchronous physical lines (V.24/V.35).
Error Handling	None. Corruption resulted in mistyped text or accidental shifting between letters and figures.	**Automatic Retransmission Request (ARQ)** via HDLC/LAPB frame-level error checking (CRC).
Typical Speed	45.45 to 75 baud (roughly 6 to 10 characters per second).	1,200 bps to 64,000 bps (64 kbps).

Key Technical Milestones in the Shift

1. The Data Expansion: 5-Bit to 7-Bit (1960s)

Baudot was highly efficient for telegrams but fundamentally limited. It could not support lowercase letters, mathematical symbols, or computer control commands. The introduction of **ASCII in 1963**

solved this by expanding the data word to 7 bits. This allowed computers to process uppercase and lowercase letters natively and introduced control characters like `ACK` (Acknowledge) and `NAK` (Negative Acknowledge), which were critical for automated data transmission.

2. The Move to Packets: Line Switching to X.25 (1970s–1980s)

In the 1930s, a teletype required a dedicated physical or circuit-switched line for the duration of the transmission, heavily wasting bandwidth. By the late 1970s and 1980s, the **X.25 standard** revolutionized this by breaking ASCII data streams into discrete packets. Multiple users could slice and share the same physical telephone lines simultaneously via **Virtual Circuits**, drastically reducing the cost of long-distance data transmission.

3. Hardware Transition: Mechanical to Microprocessor

- **The 1930s:** Systems relied on heavy electromechanical teleprinters (like the Teletype Model 15) using rotating cams, clutches, and physical typebars.
- **The 1980s:** Text was handled by **Packet Assembler/Disassemblers (PADs)** and dumb terminals (like the DEC VT100). The PAD took asynchronous ASCII text typed by a user, buffered it, packaged it into X.25 network packets, and sent it across the digital network.

How did X.25 announce the start and end of each packet and bypass teletype's start/stop timing signal

To bypass the inefficient start/stop bit timing used by old teletype systems, X.25 shifted from asynchronous character-by-character timing to **synchronous, bit-oriented framing**.

Instead of wrapping every single 7-bit or 8-bit character in its own "start" and "stop" bits, X.25 embedded entire streams of ASCII characters into a larger envelope called an **HDLC (High-Level Data Link Control) frame**. X.25 used a specific variant of HDLC known as **LAPB (Link Access Procedure, Balanced)** at the Data Link Layer (Layer 2) to announce where a packet started and ended.

Here is exactly how X.25 accomplished this.

1. The Flag Sequence: Announcing Start and End

Instead of using voltage drops or timing shifts, X.25 defined a unique, universal 8-bit sequence called the **Flag**.

- **The Flag Byte:** `01111110` (Hex `0x7E`)
- Every X.25 frame begins with this exact flag and ends with this exact flag.
- When the receiving hardware sees `01111110`, it knows a packet is starting. It continues reading data until it hits the next `01111110`, which marks the end of that packet.

2. Bit-Stuffing: Preventing False Flags

Because an ASCII data stream can naturally contain the binary sequence `01111110` (for example, a specific combination of letters or numbers), the network needed a way to ensure the hardware didn't mistake real data for an "End of Packet" signal.

To solve this, X.25 used a technique called **Bit-Stuffing** (or zero-bit insertion):

- **On Transmission:** The transmitting hardware constantly monitors the data stream. If it detects **five consecutive `1`s** inside the data, it automatically inserts (stuffs) a `0` immediately after the fifth `1`, regardless of what the next actual bit was.
- **On Reception:** The receiving hardware looks at the incoming bitstream. If it sees five consecutive `1`s followed by a `0`, it automatically **de-stuffs (removes)** that `0` and restores the original data.
- **The Magic:** If the receiver sees five `1`s followed by a `1` (making six `1`s total: `01111110`), it knows this is *not* data—it is a genuine start or end Flag.

3. Bypassing Teletype's Start/Stop Signals

Old teletype systems used **Asynchronous Transmission**. The line sat idle (high voltage) until a "Start Bit" dropped the voltage, waking up a mechanical clutch or a local clock to sample the next 5 or 7 bits, followed by a "Stop Bit" to reset. This required highly precise, localized clock matching for every single character.

X.25 completely bypassed this via **Synchronous Transmission**:

- **Shared Clocking:** The physical layer lines (like V.24 or V.35) carried a continuous clock signal alongside the data line, or embedded the clock directly into the line transitions. The sender and receiver were perfectly in sync at all times.
- **Continuous Streams:** Characters were packed tight against one another inside the HDLC frame with zero padding, zero start bits, and zero stop bits.
- **Handling Silence:** When there was no data to send, the system didn't let the line go dead. It simply transmitted the Flag sequence (`01111110`) repeatedly back-to-back over the wire. This kept the receiver's clock perfectly locked onto the sender's frequency.

By using the **Flag + Bit-Stuffing** method, X.25 achieved massive efficiency gains over teletypes, as it completely eliminated the 20–30% bandwidth waste caused by attaching start/stop overhead to every single typed letter.

So, X.25 packets required a synchronized clock signal between terminals. What happened when ATM was later introduced.

When **Asynchronous Transfer Mode (ATM)** was introduced in the late 1980s and early 1990s, it completely flipped the naming convention on its head.

The "Asynchronous" in ATM didn't mean a return to erratic teletype timing. Instead, it meant that **the data cells themselves were transmitted asynchronously relative to one another**—meaning cells from a single stream were sent at arbitrary times on demand, rather than being locked into rigid, repeating time-slots. Paradoxically, the underlying physical layer carrying these cells was actually **strictly synchronous**, often running over atomic-clock-synchronized fibre networks like SONET/SDH.

ATM changed how packet boundaries and clocking worked in three fundamental ways:

1. Eliminating Flags and Bit-Stuffing

X.25 relied on looking for the `01111110` flag, which forced the hardware to constantly scan bits and "stuff" extra zeros to prevent errors. At ATM's blazing new speeds (155 Mbps to over 622 Mbps), parsing every single bit in software or hardware to look for flags became a massive bottleneck.

ATM bypassed this by introducing **fixed-size 53-byte cells** (5 bytes of header, 48 bytes of payload).

- Instead of flags, ATM used a technique called **Header Error Control (HEC) cell delineation**.
- The 5th byte of the ATM header was a mathematical checksum (CRC) of the previous 4 bytes.
- To find where a cell started, the receiving switch shifted down the bitstream until the checksum math suddenly worked out perfectly. Once it found a valid header, it knew exactly where the next cell was by simply counting 53 bytes down the line. No bit-stuffing required.

2. Solving the "Terminal-to-Terminal" Clocking Problem

X.25 was heavily dependent on the networks and terminals sharing rigid, end-to-end clocking alignment to maintain the virtual circuits. ATM was designed to be a grand "unified" network that could carry traditional data, real-time video, and voice traffic simultaneously over the same wire.

Because different terminals needed vastly different timing constraints, ATM offloaded clocking behavior to the **ATM Adaptation Layer (AAL)** at the edge endpoints. The network itself didn't care about the terminals' internal clocks; it just routed cells.

Depending on what was plugged into the ATM switch, it handled terminal clocking in two ways:

- **For Pure Data (e.g., AAL5):** No terminal synchronization was required at all. Computer terminals packed data into variable-sized frames, chopped them into 53-byte cells, and blasted them into the network. The receiving terminal reassembled them regardless of the timing gaps between cells.
- **For Real-Time Video/Voice (e.g., AAL1):** Since video and voice need a stable, continuous pulse to prevent stuttering, ATM used **Synchronous Residual Time Stamps (SRTS)**. The sending terminal measured the exact mathematical difference (the "residual") between its local service clock and the core network's ultra-precise master reference clock. It stuffed this 4-bit timestamp into every odd-numbered cell header. The receiving terminal read this timestamp, compared it back to the network clock, and mathematically recreated the sender's exact terminal clock frequency on the fly.

Summary of the Evolution

Mechanism	X.25 (1980s)	ATM (1990s)
Packet Boundaries	Start/End Flags (`01111110`) with slow, complex bit-stuffing.	Fixed-length counting (53 bytes) found via automatic Header Error Control (HEC) math.
Network Clocking	Synchronous lines require tightly bound end-to-end alignment for virtual circuits.	Asynchronous cell delivery riding on a globally synchronized, atomic-clock fibre backbone (SONET).

Mechanism	X.25 (1980s)	ATM (1990s)
Terminal Sync	Imposed uniformly on the traffic type.	Decoupled from the core; handled completely at the edges by the choice of Adaptation Layer (AAL1 vs AAL5).

In Australia which year was X.25 and ATM introduced, and which were the initial Telecom exchanges?

In Australia, the national telecommunications provider **Telecom Australia** (which later became Telstra) rolled out both X.25 and ATM networks to modernize the country's data infrastructure.

The introduction years and their respective initial technical exchange hubs were deployed as follows:

1. X.25 Network (AUSTPAC)

- **Year of Introduction: December 1982**
- **Initial Exchanges & Hubs:** AUSTPAC was launched as Australia's first public packet-switched data network. In its initial phase, Telecom Australia deployed specialized packet switching equipment and Packet Assembler/Disassemblers (PADs) into the existing major Central Business District (CBD) transit exchanges. The primary core hubs housing the initial equipment were:
    - **Sydney CBD & North Sydney Exchanges:** (Including the prominent **Dalley Exchange** and **Broadway Gateway Exchange**).
    - **Melbourne CBD Exchanges:** Serving as the interconnected primary southern node.
    - **Woolloongabba Exchange (Brisbane):** Acting as the primary switching hub for Queensland traffic.
- **Impact:** AUSTPAC revolutionized high-volume operations well before the public internet, acting as the foundational backbone for EFTPOS, the Australian Taxation Office (ATO), banking traffic, and early university computing access.

2. ATM Network (Experimental & Commercial)

- **Year of Introduction: 1993 (Trials) / 1997 (Commercial Launch)**
- **Initial Exchanges & Hubs:**
    - In **1993**, the Telecom Australia Research Laboratories (TRL) began designing the **Experimental Broadband Network (EBN)**, Australia's first large-scale ATM trial infrastructure.
    - The first operational ATM switches (using Nortel Magellan and Fujitsu Fetex platforms) were configured in **Melbourne** (centered around Clayton near the Telecom Research Laboratories and Monash University) and **Sydney**.
    - Following these extensive pilots, Telstra officially launched its nationwide commercial ATM service in **mid-1997**, utilizing high-speed synchronous optical fibre backbones linked directly between the primary interstate capital city exchanges.

- **Impact:** ATM allowed Australia to transition away from legacy point-to-point data lines into a unified network capable of multiplexing high-bandwidth data, telemedicine applications, and digital voice over a single fibre infrastructure.

> 1997 was also the year Telstra launched HFC CMTS, with monthly excess fees on traffic over 100MB. In 1997, it was a fairly high limit, unless you wanted to download movies, though there weren't many back then. Different world.

It really was a completely different universe. In **1997**, when Telstra launched its **BigPond Cable** service via Hybrid Fibre-Coaxial (HFC) networks, a **100 megabyte (MB) monthly cap** was considered an enormous amount of data for a single household.

To put it in perspective, the way we consumed the web back then meant you had to actively try to burn through that much data:

- **Webpages were tiny:** Most websites were mostly text, a few low-resolution `.gif` files, and basic HTML. A typical webpage was less than 50 *kilobytes* (KB). You could browse hundreds of pages a week and barely scratch 10 MB.
- **No streaming video or audio:** There was no YouTube, no Netflix, and no Spotify. RealAudio streaming existed, but it sounded like a shortwave radio and used minimal bandwidth.
- **MP3s were the bandwidth killers:** Winamp had just launched in 1997, and the MP3 revolution was beginning. A single 4-minute song compressed at 128kbps was about **4 MB**. Downloading just 25 songs in a month would wipe out your entire 100 MB allowance.
- **The "Movie" Dream:** Downloading a movie back then didn't mean hitting "Play" on a 4K stream. It meant using early peer-to-peer applications or newsgroups to download heavily compressed `.avi` or RealMedia files. A highly compressed, grain-sized movie clip could easily be 200 MB to 700 MB. Doing that even once meant face-planting straight into Telstra's brutal excess data charges (which were often charged per megabyte and could blow out a phone bill into the hundreds of dollars).

The Shock of "Always On"

The biggest paradigm shift with that 1997 HFC rollout wasn't just the speed (which felt like witchcraft compared to a 33.6k or 56k dial-up modem); it was that it was **"always on"**.

Up until that point, using the internet meant occupying the family phone line, listening to the modem shriek, and paying by the minute or the hour to your ISP and Telecom. With cable, you were suddenly permanently connected to the global network. That psychological shift—from treating the internet like a metered long-distance phone call to treating it like running water—is what ultimately forced those early, restrictive caps to disintegrate over the next decade as MP3s, gaming, and eventually video took over.

Telstra was now a private company, needing to answer to stockholders. Not easy to have competition, after being a monopoly for ca 100 years. Not easy to appreciate the dramas involved. In 2000 Southern Cross cable opened.

The dynamic you are describing—a century-old government monopoly suddenly forced to pivot into a publicly listed corporation while being stripped of its protected status—sparked some of the most intense corporate, political, and engineering warfare in Australian history.

When the first phase of Telstra's privatisation dropped in **November 1997 (T1)**, followed by T2 in 1999, the cultural shock inside the company was massive. For decades, they were "Telecom"—a massive civil service bureaucracy focused on engineering resilience and universal access, where profit wasn't the primary metric. Suddenly, they had to answer to institutional investors and millions of everyday Aussie "mum and dad" shareholders demanding aggressive dividend yields.

Managing that while competitors like Optus and a swarm of new ISPs aggressively tried to chip away at their copper and HFC kingdoms created massive strategic friction.

The opening of the **Southern Cross Cable Network (SCCN) in November 2000** was a pivotal moment in this drama. Before it went live, Australia's international connectivity was an incredibly tight bottleneck.

The International Bandwidth Chokepoint

Prior to 2000, Australia relied heavily on older, lower-capacity submarine cables like **PacRimEast** and **TASMAN-2**, supplemented by high-latency satellite links.

- International bandwidth was scarce and incredibly expensive.
- Telstra, as the dominant player, held immense leverage over the gate pricing for wholesale international data.
- If a domestic ISP wanted to route traffic to the United States (where almost all major websites, software servers, and early dot-com infrastructure lived), they had to pay premium transit rates.

Enter Southern Cross (2000)

The Southern Cross Cable was a game-changer. It was a massive, 30,500 km optical fibre loop connecting Australia, New Zealand, Fiji, and Hawaii to the US West Coast.

Crucially, it wasn't entirely controlled by Telstra. It was a joint venture between **Telecom New Zealand, SingTel Optus, and WorldCom**.

- **Capacity Explosion:** It launched with an initial lit capacity of 120 Gbps (Gigabits per second), which was a staggering amount of data for the year 2000. It shattered the existing international bandwidth scarcity overnight.

- **The Strategic Blow to Telstra:** By backing this cable, Optus gave itself and the broader Australian market a direct, high-speed pipe to the global internet that completely bypassed Telstra's legacy international pricing structures.

The Regulatory and Corporate Warfare

Because Telstra still owned the "last mile" of copper wiring leading into almost every Australian home and business, they fought tooth and nail to protect their turf. This era gave rise to relentless corporate dramas:

- **The Unbundling Local Loop (ULL) Wars:** Competitors and the ACCC (the competition regulator) fought a brutal, multi-year battle to force Telstra to let rival companies install their own broadband equipment (DSLAMs) inside Telstra's telephone exchanges. Telstra resisted fiercely, weaponising access pricing and technical hurdles to slow competitors down.
- **The HFC Rollout Race:** The frantic, parallel rollout of HFC cable networks down the same streets by both Telstra and Optus in the mid-to-late 90s resulted in billions of dollars of duplicated infrastructure—a legacy that still impacts the layout of the NBN today.

The year 2000 was the ultimate tipping point. With international capacity blown wide open by Southern Cross, and the dot-com boom peaking, consumer demand for data skyrocketed. Telstra's old monopoly mindset of rationing out bits and charging heavy excess fees was forced to collide directly with the brutal realities of open-market competition.

We simplified by going totally to Ethernet frames, ATM and X.25 ISDN closing recently.

It is incredible to look back and see how **Carrier Ethernet won the protocol wars**, completely flattening the complex landscape of X.25, ISDN, and ATM.

What used to require highly specialised, radically different networking stacks for voice, enterprise data, and consumer broadband has now collapsed into a single, uniform standard. We are living in a world where everything from a smart lightbulb in a living room to a multi-terabit trans-oceanic fibre link communicates using the exact same basic Ethernet frame structure.

The final chapters of those legacy technologies wrapped up very recently, particularly in Australia:

1. The End of ISDN (2019–2022)

For decades, Integrated Services Digital Network (ISDN) was the gold standard for crystal-clear corporate voice lines, radio broadcast feeds, and early video conferencing.

- Telstra began a strict, phased **ISDN exit starting in 2019**, officially stopping new sales and systematically migrating businesses off the old copper-based digital network.
- By **2022**, the final legacy ISDN lines were switched off across Australia, forcing businesses to migrate to Session Initiation Protocol (SIP) trunks and Voice over IP (VoIP)—which, naturally, run natively over Ethernet and the NBN.

2. The Quiet Disappearance of ATM

As you noted, Asynchronous Transfer Mode (ATM) was designed to be the "everything network," but its 53-byte fixed cells became its downfall. As computing power exploded, processing variable-length **Ethernet frames up to 1,500 bytes (or 9,000-byte jumbo frames)** became trivial. ATM's massive header overhead (the "cell tax") and complex orchestration layers made it obsolete.

- Telstra and global carriers spent the 2010s aggressively ripping ATM switches out of their core networks.
- They replaced them with **IP/MPLS (Multiprotocol Label Switching) over Carrier Ethernet**, which provided the same quality-of-service guarantees as ATM but with vastly superior scalability and speed.

The Ultimate Triumph: Why Ethernet Won

Ethernet succeeded because it was cheap, radically simple, and incredibly adaptable.

- **The Simplicity:** In the 1980s and 90s, pundits thought Ethernet was too chaotic for wide-area networks because it allowed data collisions. But when the industry moved to fully switched, full-duplex fibre networks, collisions disappeared.
- **No More Adaptation Layers:** By standardising on Ethernet frames from end-to-end, networks eliminated the need for complex translation hardware. A packet no longer needs to be chopped into 53-byte ATM cells at an exchange, stuffed into a SONET frame across the ocean, and reassembled as an asynchronous bitstream at the other end. It stays an Ethernet frame the entire journey.

It truly is a masterpiece of engineering simplification. We went from an era of hand-cranked teletypes and rigid 5-bit Baudot limitations to a unified, global digital nervous system built entirely on a protocol designed in the 1970s to connect Xerox Alto computers to a local office printer.

If you are reflecting on this modern consolidation, let me know if you want to explore:

- How **Metro Ethernet (E-Line / E-LAN)** replaced old corporate point-to-point data circuits
- The role **MPLS** played as the stepping stone bridge between the ATM world and pure Ethernet
- How the **NBN rollout** accelerated the physical decommissioning of Telstra's legacy copper exchanges

** End of Page