Read or listen to this article online

Scan the QR code to open this article on XO.PL

Article source:

https://xo.pl/it-made-fun-net-osi

NETWORKING

The OSI model: seven floors of bureaucracy on the Packet Highway

A new government office opened on the Packet Highway. It was no ordinary office. It had seven floors, two identical buildings — one for the sender and one for the receiver — and a huge sign: “OSI MODEL. NO ELEVATOR. EVERY DEPARTMENT DOES ITS OWN JOB.”

The Manager arrived because he had heard the word “model” and assumed someone had ordered a new company car. Grzesiu arrived because he had heard the word “network” and correctly suspected somebody was about to unplug a cable. Agnieszka came with one simple task: send a message asking about an invoice.

“WHY SEVEN FLOORS?” demanded the Manager.

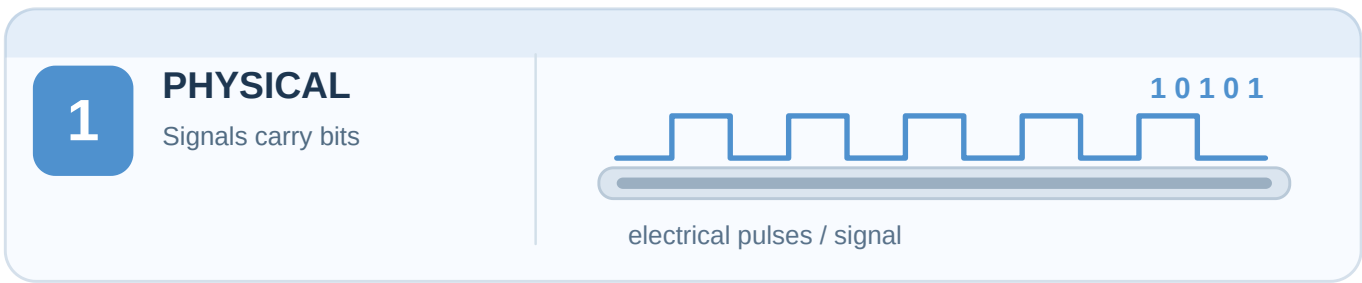
“Because each layer solves a different problem,” Grzesiu replied.

“AND THE EIGHTH?”

“We'll discuss the eighth at the end. Unfortunately, I already know its director.”

The OSI (Open Systems Interconnection) model is a seven-layer reference model describing the functions needed for network communication. It is not seven mandatory programs that you must install, nor an exact construction plan for the Internet. Instead, it helps explain who handles signals, local connections, routes, transport, and application services. We'll start with the foundations: layer one.

Floor one — Physical. No signal, no party



Electricians, fiber installers and radio specialists worked on the ground floor. The Manager walked in, looked around at the cables and immediately found a power strip.

“I'LL UNPLUG THAT. IT'S IN THE WAY!”

“Please don't,” said Grzesiu. “Without this layer, everybody upstairs will have to send messages by carrier pigeon.”

Layer 1 — Physical carries bits as signals over a defined medium: copper cable, optical fiber or radio waves. It concerns signaling methods, electrical or optical characteristics, connectors and other technology-specific requirements. A bit is a logical zero or one; tiny cardboard zeros and ones do not actually fly down a cable. Real physical signals do.

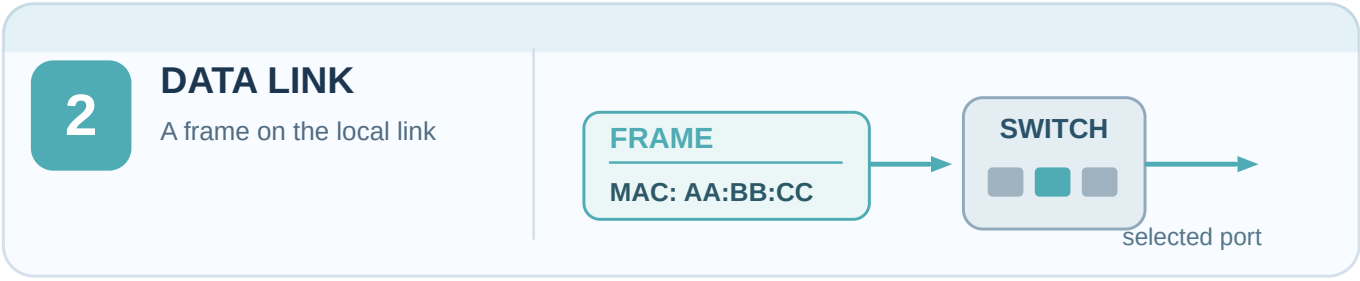
Imagine two devices connected by Ethernet. If the cable is damaged or the interface cannot establish a link, no amount of clever work at the upper layers will conjure up a missing signal. Conversely, a lit link indicator does not prove that an application works. It only tells you something about the link.

“WHAT IF I WRITE ‘INTERNET’ ON THE CABLE? WILL IT GO FASTER?”

“No. But if you write ‘DO NOT UNPLUG’, it may keep working longer.”

The floor notice reads: **unit — bits; question — can the signal travel at all?** The Physical layer does not know IP addresses, website names or port numbers. Its job is to carry the signal, not read the letter.

Floor two — Data Link. Who is at the other end of this stretch?



Upstairs, employees put documents into frames. Each frame carried the information needed to cross the current section of the network. The Manager demanded that every frame be marked “Kraków”.

“We're working on a local link,” an employee pointed out. “We need to know where to hand the frame *now*, not just the final destination.”

Layer 2 — Data Link organizes the exchange of **frames** across a particular link or local communication domain. In Ethernet, this includes MAC addresses, frame switching and error-detection mechanisms such as the FCS. A checksum helps detect damaged frames, but it does not by itself guarantee retransmission and delivery.

Grzesiu put three building models on the table: Agnieszka's computer, a switch and a router.

“The computer wants to send data to a server outside its own network. Which destination MAC address goes into the Ethernet frame on the first link?”

“THE SERVER'S, OBVIOUSLY!” announced the Manager.

“Usually the MAC address of the next device on that local segment, such as the default gateway. The server's destination IP address remains in the packet. When a router forwards that packet over another link, it uses a suitable new frame.”

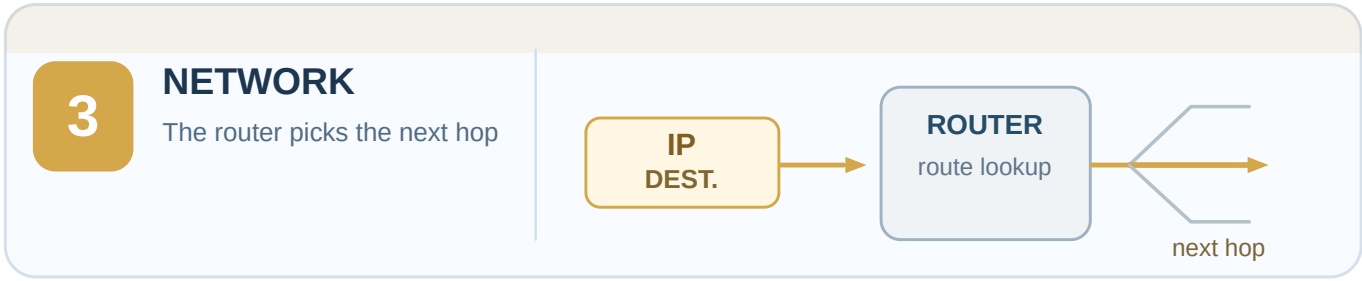
This distinction matters: **a layer-2 frame serves the current link, while a layer-3 IP packet may cross many links.** Mechanisms such as ARP for IPv4 and Neighbor Discovery for IPv6 help discover information about neighbors; they do not make MAC addressing into a worldwide routing system.

“So the switch doesn't need the address of a hotel across the planet?”

“No. Otherwise you'd make it choose routes based on the color of the envelope.”

The floor notice reads: **unit — frame; question — how do we pass data across this particular link?**

Floor three — Network. Knowing the destination is not knowing the next turn



A huge map of the Packet Highway covered the third floor. Hundreds of roads, cities, detours and signposts. A dispatcher sat beside a routing table. The Manager pointed toward Kraków.

“I KNOW WHERE WE'RE GOING. DRIVE STRAIGHT!”

“The bridge is missing,” said the dispatcher.

“THEN DRIVE STRAIGHTER!”

Layer 3 — Network uses logical addresses, such as IPv4 and IPv6, and enables packets to be forwarded between networks. A router examines the destination IP address and uses its routing table to choose the next hop or the appropriate outgoing interface. It does not have to know the packet's entire future journey; other routers make their own decisions along the way.

For example, Agnieszka's laptop uses an address on her home network while the server is elsewhere. The local switch does not chart a route across the Internet. The packet goes to a gateway and onward through other routers. The MAC addresses used on individual links may change, while the destination IP address ordinarily remains the final target. Address translation and tunneling are exceptions, not the basic rule of IP forwarding.

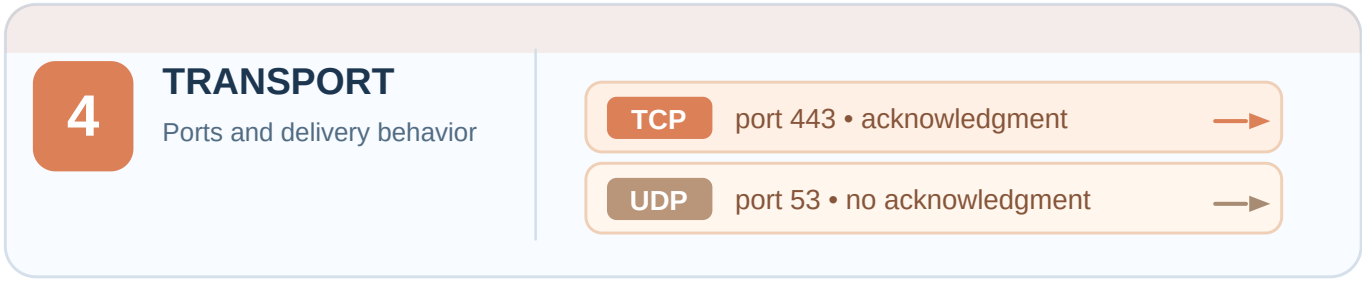
“WHY DOESN'T THE PACKET ALWAYS TAKE THE SHORTEST ROAD?”

“Routes depend on available information, routing policy and network parameters. The visually shortest road may not be chosen. We explain this in our **routing story**.”

Layer 3 does not guarantee delivery, choose the receiving application or repair a broken cable. IPv4 and IPv6 also include mechanisms to prevent packets from circulating indefinitely.

The floor notice reads: **unit — packet; question — where is the logical destination, and what is the next hop?** For logical addresses, see our **IP address story**.

Floor four — Transport. The building is correct, but which desk?



Two delivery companies worked on the fourth floor: TCP and UDP, familiar to our regular readers. They stood before a pile of shipments. All were headed to the same building, but different departments.

“WE HAVE AN IP ADDRESS!” the Manager protested. “WHAT MORE DO YOU WANT?”

“IP identifies the device,” said Grzesiu. “That device may run a browser, a web server, a mail client and many other programs at once. We need to distinguish their conversations.”

Layer 4 — Transport supports communication between the appropriate service endpoints. TCP and UDP use **port numbers**, among other fields. IP addresses, protocol and port information help distinguish exchanges or flows. Don't confuse a TCP or UDP port with a physical socket on a switch: same word, different job.

The impeccably dressed TCP courier raised a form.

“With TCP we establish a connection. I look after byte ordering, acknowledgments, retransmissions, flow control and congestion response.”

The UDP courier was already halfway out of the door.

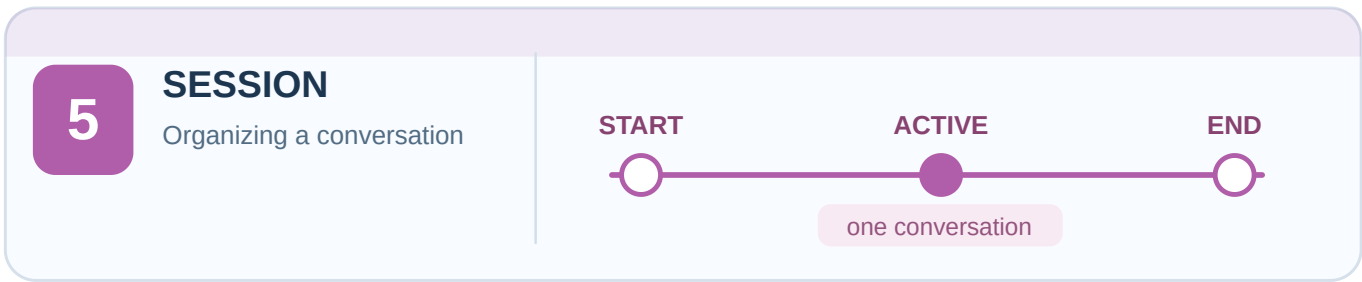
“I send a datagram. UDP itself offers no delivery, ordering or automatic retransmission guarantee. If an application needs more, it can arrange it elsewhere.”

“SO UDP IS BROKEN?”

“No,” said Grzesiu. “It's simpler and meets different needs. Not every message needs registered mail, and modern protocols can build reliability on top of UDP. We have a separate **TCP and UDP story**.”

The floor notice reads: **unit — TCP segment or UDP datagram; question — which conversation is this, and how should data be transported?** Transport does not route packets through the Internet or need to understand the contents of a web page.

Floor five — Session. Who is talking, and is the conversation still going?



The fifth floor contained meeting rooms. A receptionist managed conversation schedules, participant lists and a stamp reading “FINISHED”. The Manager burst in without knocking.

“END ALL SESSIONS IMMEDIATELY! THINK OF THE PAPER WE'LL SAVE!”

“Not every conversation ends after one sentence,” the receptionist sighed.

Layer 5 — Session describes organizing dialogue between cooperating applications: establishing, maintaining, synchronizing and terminating sessions as required by the service. Imagine a multi-stage exchange in which both parties need to know that they are still in the same conversation and perhaps where to resume it.

“Is a session just a TCP connection?” asked Agnieszka.

“No. TCP has its own connection state, but OSI session functions are conceptually different. One application-level session may span several transport connections, and communication over UDP can also maintain its own state. Don't equate every TCP connection with layer 5.”

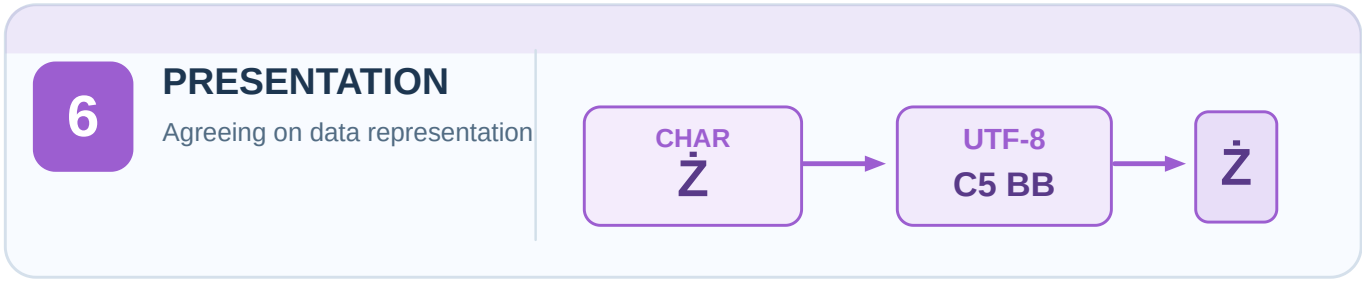
In the real Internet there may be no separate protocol or program labeled “session layer”. Session responsibilities can be implemented in applications and libraries. This is one of the places where forcing every modern implementation into seven completely independent boxes becomes misleading.

“THEN WHY DO WE HAVE A DEPARTMENT IF IT DOESN'T ALWAYS HAVE ITS OWN COUNTER?”

“Because we're separating responsibilities to understand them. A real company can perform several responsibilities in one office.”

The floor notice reads: **question — how is the conversation organized and maintained?**

Floor six — Presentation. Please use a format the recipient understands



Translators, editors and data-packaging specialists occupied the sixth floor. The Manager brought a card bearing the Polish tongue-twister “Zażółć gęślą jaźń” and confidently declared that any computer would understand it.

“Which character encoding shall we use?” asked the translator.

“POLISH!”

“That isn't a character encoding, Manager.”

Layer 6 — Presentation describes representing and transforming information so both parties can interpret it. This includes data formats, character encodings, serialization, compression and, in some conceptual mappings, encryption and decryption. Text encoded as UTF-8 is not the same as bytes interpreted using an

incompatible character set. A JPEG file will not magically become a text document because someone gives it an attractive filename.

Agnieszka handed the clerk a document.

“Prepare it so the other side doesn't have to guess whether it's text, a photograph or a compressed file.”

“Certainly. And if encryption is involved, the receiving party also needs an appropriate way to decrypt the information.”

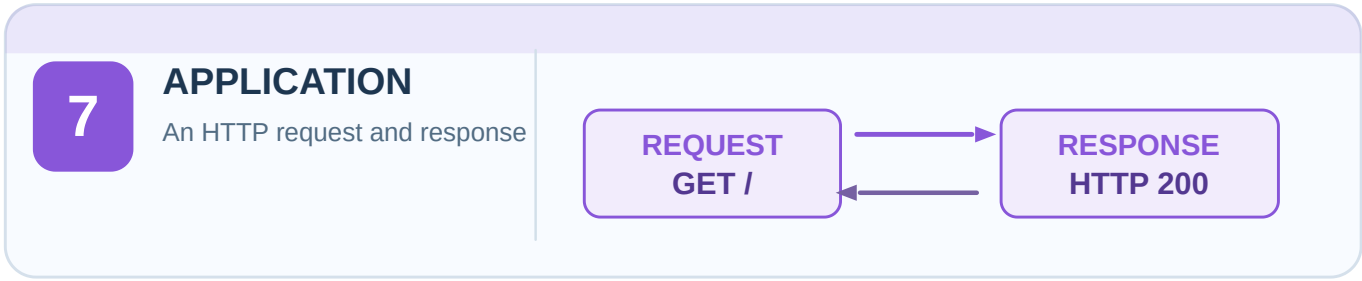
An important qualification: in the common TCP/IP stack, data formatting and TLS do not simply form one mandatory, separate “layer 6”. These functions help illustrate presentation-layer responsibilities, but real implementations organize them differently.

“IF I SEND GIBBERISH, WILL THE NETWORK FIX IT?”

“No. The network might deliver that gibberish perfectly. The recipient still won't understand it.”

The floor notice reads: **question — in what form should the information be represented?**

Floor seven — Application. At last, somebody knows why we're here



The seventh floor contained a service counter. Agnieszka opened the invoice website. The Manager rushed up to the clerk.

“SO THIS IS WHERE YOU KEEP THE ENTIRE INTERNET?”

“No. This is where applications use network services and protocols. We do not keep the Internet in a cupboard.”

Layer 7 — Application describes the network services and protocols used by applications. Examples include **HTTP** for website requests and responses, **DNS** for information about names, and email-related protocols. A person sees the browser window, but a window is not the definition of the Application layer — the important part is the protocol used for communication.

“I want to open my invoice,” Agnieszka said.

“Then the application prepares the relevant request. The lower functions provide communication, but they do not decide what an HTTP request means.”

“HOW DOES THE BROWSER KNOW WHERE THE SERVER IS?”

“Among other things, through naming mechanisms such as **DNS**. Then all the floors we've already visited become useful.”

The floor notice reads: **question — what network service does the application need, and which protocol does it use?**

Two buildings, one message. Take the stairs!

After inspecting all seven floors, Agnieszka finally decided to send her invoice request. The Manager spotted a large arrow between the buildings and immediately demanded an express elevator from the seventh floor to the road.

“WHY WALK DOWN SIX FLOORS? BUILD A SHORTCUT!”

“No elevator in our model,” said Grzesiu. “Each level has its own job.”

On the **sender's side**, a service request starts at layer 7. Representation and dialogue functions apply where needed (6 and 5). Transport prepares its exchange, for example a TCP segment containing port numbers (4). The Network layer makes an IP packet (3), Data Link puts it in a suitable frame for the current link (2), and the Physical layer carries signals (1).

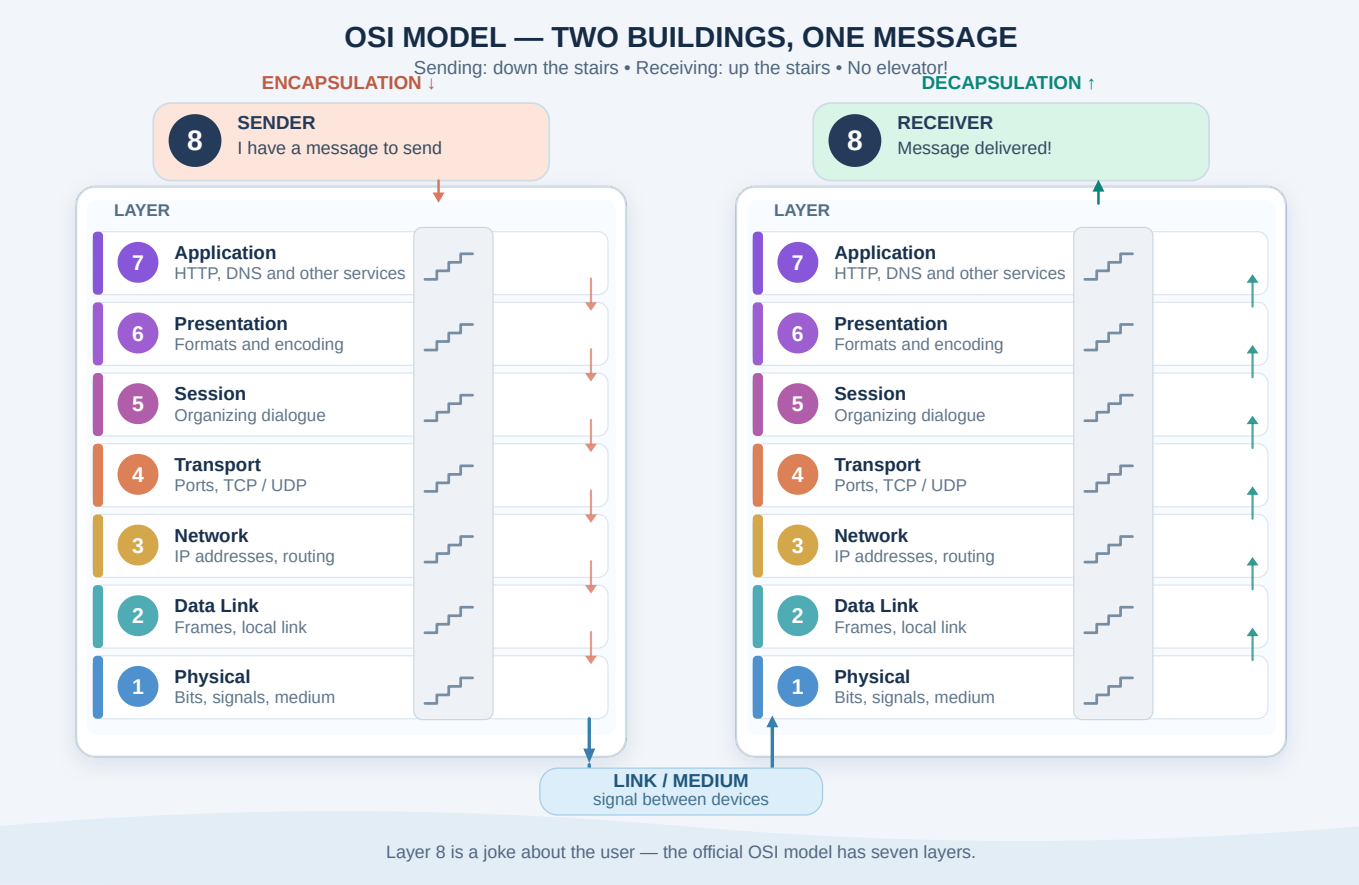
Along the way, a router does not need to read the invoice number or climb to floor seven. An ordinary IP router primarily uses layer-1-to-3 functions on its interfaces: receive a frame, examine and forward the IP packet, choose a next hop and send it in a suitable new frame. Some network devices also implement higher-layer services, but that's a different story.

At the **receiver's side**, processing goes the other way: signal (1), frame (2), IP packet (3), appropriate transport communication (4), session and data interpretation functions as applicable (5 and 6), and application protocol handling (7). The server prepares a reply and sends it back. In practice, sender and receiver are not permanent identities: both sides can send and receive.

This is a simplified map of responsibilities, not a compulsory tour through seven separate programs. The key principle is **encapsulation**: as data goes down, lower layers add the information needed for their own work. On arrival, relevant headers and other information are processed and removed through **decapsulation**. That's how an HTTP request can cross many different networks without every switch having to understand a web page.

“CAN WE AT LEAST THROW THE PACKET OUT OF A SEVENTH-FLOOR WINDOW?”

“Certainly,” said Grzesiu. “But then this becomes a very short article about damaged cargo.”



What can the floor number tell us when something breaks?

Suddenly Agnieszka reported that her invoice page would not open. The Manager immediately announced:

“THE INTERNET IS BROKEN! REPLACE THE WHOLE INTERNET!”

“Let's ask questions,” said Grzesiu. “Is there a physical signal and a link? Can the device reach its local gateway? Is the IP configuration correct and is there a route to the destination? Is the relevant port and service reachable? Does DNS answer? Is the server returning an application error?”

This is the practical power of the OSI model: **it helps organize troubleshooting by the kind of failure.** A missing physical link, a local frame-delivery problem, an absent IP route, an unreachable port and an HTTP application error are not the same fault, even if the user describes each as “It doesn't work”.

It is not, however, a mandatory checklist that every experienced engineer must execute from bottom to top. A useful diagnosis starts where the symptoms suggest. The model provides a shared vocabulary, not instructions for pressing every button in order.

“I HAVE A DIAGNOSIS TOO!” shouted the Manager. “THAT CABLE AGAIN!”

“This time the server is reporting a permissions error.”

“THE CABLE DOESN'T HAVE PERMISSION?”

Layer eight — the director's office that wasn't on the plans

After the problem was fixed, Agnieszka received the server's reply. All seven floors had done their work. The signal arrived, frames were passed along, routers selected routes, transport delivered the data and the application displayed the message.

A sign lit up on the receiver's roof: “LAYER 8 — HAPPY RECIPIENT”.

The Manager narrowed his eyes.

“SO THERE IS AN EIGHTH FLOOR! THAT'S MY OFFICE!”

“No,” said Grzesiu. “The official OSI model has seven layers. ‘Layer 8’ is a joke about the user — the human factor. It can make brilliant use of technology or turn a working service into a helpdesk ticket with one click.”

The Manager read the screen. The reply said: “Invoice available after signing in.”

“NOTHING ARRIVED!”

“It did. You need to sign in.”

“BUT I DON'T WANT TO SIGN IN!”

“Then the network has done its job, and we're back at layer eight.”

The main lesson: seven layers help us understand seven categories of communication tasks. Layer one carries signals, two handles the current link, three handles logical addresses and routing, four transports between services, five organizes dialogue, six represents information, and seven provides application-level protocols. In the real Internet these functions do not always have separate offices or perfectly sharp borders. That is precisely why the model is useful: it helps explain *what* must happen without pretending every network is built like the same building.

Central Packet Handling Office notice: both buildings are operating normally. The stairs are clear, there is no elevator, and no emergency exit between layers has been found. The Manager has ordered a sign reading “LAYER 8 — MANAGEMENT”. Grzesiu has ordered a sign reading “DO NOT TOUCH”. Guess which one proved more useful. 🚦🤣❤️