PART A – OSI MODEL THEORY

1.OSI Layer Explanations

Layer 1 – Physical Layer

The Physical Layer deals with how bits move across a network cable. Rather than focusing on data meaning, it pays attention to physical pieces - things like cables, plugs, jacks, signal power, or even the internal card connecting your device. At this level, info exists as electrical signals or light bursts - just raw pulses, no interpretation. The key point? Devices must be correctly connected so they can send untouched digital data between each other. Imagine a path where vehicles roll - no matter what they carry, the trail simply allows passage. Stuff such as Ethernet cables, fiber optics, or wireless signals takes care of this layer instead.

Layer 2 – Data Link Layer

The Data Link Layer takes raw signals from the physical level, turns them into pieces called frames. Rather than IP addresses, it uses MAC ones to tell devices apart on a local setup. On top of handling who gets access to cables or wireless channels, it catches errors in data, sorts out delivery hiccups within one LAN. What's its core task? Sending info to the correct device nearby. Stuff such as Ethernet or Wi-Fi works here. Imagine it like a home address - after you're on the right road, that number gets your letter to the exact entrance.

Layer 3 – Network Layer

The Network Layer manages how information is tagged and routed, letting data travel between different networks. Yet it splits info into pieces, attaching an identifier - say, an IP address - so devices anywhere can connect. Routers at this level decide where each packet moves next, guiding flow step by step. Protocols like IPv4, IPv6, or ICMP operate here, handling tasks without fuss. Imagine it similar to a navigation app picking fast routes from you to a distant spot, shifting paths so things keep moving easily.

Layer 4 – Transport Layer

The Transport Layer manages how apps communicate across different devices. While data is large, it breaks into pieces for smoother delivery. Even if packets arrive out of order, sequencing techniques ensure they're reassembled correctly. Whenever a segment gets lost during transit, recovery mechanisms trigger a resend. TCP keeps things intact by tracking packets and confirming delivery. On the other hand, UDP ditches that process to speed up transmission - losses included. Instead of relying on chance, port numbers route data to correct apps - say, browsers or DNS tools. Imagine sending mail: TCP acts like a tracked parcel with signatures at every step. In contrast, UDP is closer to dropping a postcard somewhere, trusting it'll reach its destination.

Layer 5 – Session Layer

The Session Layer handles communication between two devices. Once gadgets begin exchanging data, this layer arranges the initial setup. While info travels both ways, it keeps the connection running smoothly. Should the link drop, recovery starts from a saved point - not from scratch. Apps such as NetBIOS or RPC operate at this level. Imagine phoning a friend - picking up the phone kicks things off, chatting keeps it alive, putting the receiver down wraps it up - all steps in a single conversation.

Layer 6 – Presentation Layer

The Presentation Layer ensures data from one device runs on another, regardless of internal storage differences. Rather than align formats head-on, it converts things like codes, symbols, or file sizes before transmission. While browsing securely via HTTPS, encryption happens at this level. What's its core task? Preserve the true meaning of information across different systems. Picture a person instantly changing Spanish into Japanese during talk - letting both sides understand clearly without mix-ups.

Layer 7 – Application Layer

The Application Layer takes care of jobs for programs you interact with - like browsing the web or checking email. Yet it's more about how those tools link up to networks and display data. Inside this layer are common systems such as HTTP, FTP, SMTP, along with DNS and DHCP. They aren't apps like Chrome or Outlook themselves, rather what makes them function online. Think of it like a store's help desk - the spot where folks ask stuff and kick things off.

2.OSI Mnemonic

Chosen Mnemonic: "Please Do Not Throw Sausage Pizza Away"

To remember the OSI model's seven layers from bottom to top, I use this phrase: "Please Do Not Throw Sausage Pizza Away." It works well since each word's first letter matches a layer, and the image is odd enough to stick. This is how each bit ties to its level, plus what that level actually does.

1. “Please” → Physical Layer (Layer 1)

The word "Please" means the Physical Layer - the base part of the OSI model. That's where sending data kicks off, right at the lowest step.

This layer deals with how gadgets link up - through cords or wireless signals. Voltage amounts, ports, cables, radio frequencies, transfer speeds, physical gear, kinds of signals, plug designs, distance between units, along with sending techniques - all fit here

Cables used in networks - either copper or fiber - are built different but work alike when moving data around

- Connectors and pins
- Voltages along with electrical impulses
- Radio waves used by Wi-Fi systems

The actual raw info - just zeroes and ones - travels across the pathway via signals rather than physical connections

This level pushes pure data - no concern for meaning whatsoever. What stays with me is "please," since that’s where actual things start happening.

2. “Do” → Data Link Layer (Layer 2)

The following word, "Do," points to the Data Link Layer - sometimes called level two.

This section manages how gadgets communicate within a single nearby network. Even though signals travel through wires safely, they still need proper formatting. Meanwhile, data stays organized the right way.

Tasks you’ll handle:

- Using MAC addresses to identify devices in a LAN
- Framing (creating data frames)
- Error detection
- Flow control
- Things such as switches operate in this spot.

The word 'Do' makes me think this level manages reliable delivery inside one system - while also linking parts together through consistent flow.

3. “Not” → Network Layer (Layer 3)

The word "Not" means Network Level - though folks sometimes call it something else.

This chunk deals with IP addresses plus routes, letting info figure out how to move across various networks while heading toward its destination - picking options that shape every stage of the journey.

Main jobs here include:

- IP addresses are set up right here - each one managed on its own, no added complications along the way
- Routers operate at this layer
- Routing algorithms determine the best path
- Packets are put together right in this spot

The tricky term "Not" makes me think this stage is about things bigger than just nearby - linking through systems rather than staying close.

4. “Throw” → Transport Layer (Layer 4)

The fourth bit, called "Throw," means the Transport Layer - joining parts by motion instead of fixed links.

This section handles entire chats between gadgets, controls data speed while keeping things steady - splitting details into pieces through seamless flow.

Essential tasks handled locally:

- Breaking down details into pieces
- Ensuring reliable delivery using TCP
- Sending info fast - though safety’s iffy - using UDP instead
- Handling retries, confirmations, or port IDs

The word 'Throw' works here because this stage deals with moving data from one end to the other - kind of like tossing a ball. Instead of complex paths, it just connects start and finish directly. Think of it as handing off an object quickly, bypassing any lag. No added layers, no waiting.

5. “Sausage” → Session Layer (Layer 5)

The word "Sausage" connects to the Session Layer - linked via roles in how it works, not only by direct ties. It's about function shaping the link, shaped through interaction points across layers.

This bit manages how two devices communicate - sharing data through connections rather than straight hooks while they respond to one another.

Things it handles:

- Opening a session
- Keeping the conversation going while you talk
- Closing it when the job’s done
- Synchronization between applications

Examples include:

- Visiting a webpage through the internet
- Video conferencing sessions
- Remote login sessions

The word 'Sausage' reminds me of this stage - each session feels pulled thin, one bit after another, almost like pieces hooked together.

6. “Pizza” → Presentation Layer (Layer 6)

The word 'Pizza' refers to the upper level - kinda like a signal carrier in the OSI model. This piece transforms information into stuff people notice, such as images or pop-ups. Instead of staying hidden, it shows up right in front of you on apps or sites. It runs without noise, shaping how things look each time you click or scroll.

It mostly deals with layout stuff, changing shapes, mixing up info, sorting it back out, packing data tight - so anyone sending or receiving understands what’s there.

Functions include:

- Converting between formats (JPEG, ASCII, MP4, etc.)
- Data encryption (like SSL/TLS)
- Data compression for efficiency

The term 'Pizza' makes you think of different flavors or forms - proof that this layer shifts formats just as pizza appears in countless versions.

7. “Away” → Application Layer (Layer 7)

The final part, "Away," means the app layer.

This is placed up high, just close to the person. But it allows actual programs to communicate across the web via a direct route.

Here’s what you’ll run into: stuff like this

- HTTP or HTTPS - this shows up when you visit sites online
- SMTP / POP3 / IMAP – for email
- FPT – used to send files
- DNS - this converts names to IP spots using lookups instead of direct links

This happens once network services pop out from deep inside - suddenly accessible to users who need them.

Why This Mnemonic Works Well

I chose the mnemonic **“Please Do Not Throw Sausage Pizza Away”** because:

- The phrase sounds dumb - but it’s stuck in your head fast
- Each word leads directly to a single stage
- Stays on track from the start all the way to step seven - no slipping off course, just steady progress without hiccups or gaps along the path
- It moves smoothly, which makes it good for quizzes, speaking tests, or explaining Wireshark bits.
- It reduces confusion between middle layers like Session, Presentation, and Application

Conclusion

In short, "Please Do Not Throw Sausage Pizza Away" makes it fast to recall the seven OSI layers step by step. Each word matches a layer - painting a simple picture of its role. Rather than wondering what's next, I rely on this cue during real tasks or when breaking down ideas. Plus, it’s useful whenever I’m tracking data flow using Wireshark.

3.OSI vs TCP/IP Model Comparison

The OSI Model alongside the TCP/IP Model show how info moves through networks - yet each one was built for separate goals. While the OSI setup splits things into seven parts, mostly helping folks learn how data flows step by step. Every level handles something unique, like sending signals over cables, locking down security, shaping data structure, or guiding packets using IPs. Thanks to this clean split, spotting hiccups gets easier - say, figuring out if trouble comes from broken wires at Level 1, messed-up device addresses at Layer 2, or faulty path choices in Layer 3. Still, you won't see it fully used in actual networks. Rather, it works more like a rough plan or point of comparison.

The TCP/IP Model works in practice - it’s built around four layers that show how devices really talk online. Since actual protocols mix tasks from multiple OSI levels, this model merges them where it makes sense. Take HTTP, DNS, or SMTP - they manage user needs, format data, and keep sessions going all by themselves, so Layers 5 through 7 become just one layer here: the Application Layer. Instead of theory, it follows what real tech does - like making sure messages arrive safely using TCP, or quickly with UDP. IP deals with sending packets to right places, while Ethernet or Wi-Fi handles getting things onto the wire - or airwaves. From home gadgets to global web traffic, nearly every current system runs on TCP/IP.

In short, the OSI Model dives deeper plus stays more abstract, whereas TCP/IP keeps it basic and built for real-world use. Although their setups aren't the same, these frameworks match up well - every OSI level links to a matching one in TCP/IP.

Mapping Table: OSI ↔ TCP/IP

OSI Layer (7 layers)	**TCP/IP Layer (4 layers)**
Application / Presentation / Session (L7–L5)	Application
Transport (L4)	Transport
Network (L3)	Internet
Data Link / Physical (L2–L1)	Network Access / Link

4.Protocol Data Units (PDUs)

Each level of the OSI model gives data a different label. Such labels make it clearer what kind of info is worked on at every step, while showing how that data shifts when traveling across networks.

OSI Layer	**PDU Name**	**Explanation (Detailed, Human-Written)**
Layer 4 – Transport	**Segment (TCP) / Datagram (UDP)**	Data is split for delivery; TCP makes segments, UDP makes datagrams.
Layer 3 – Network	**Packet**	IP information is added so routers can move data between networks.
Layer 2 – Data Link	**Frame**	MAC addresses are added so switches can deliver data in the LAN.
Layer 1 – Physical	**Bits**	Data is sent as 0s and 1s over cables or wireless signals.

5.Addressing Concepts

Networking relies on various address kinds across OSI layers. Yet each serves a unique role, guiding info accurately between locations.

1. MAC Address – Layer 2 (Data Link Layer)

A MAC address identifies each network device, given by the maker during production. This 48-bit identifier typically shows up in hex format - like A4:5E:60:1B:9F:22.

The MAC address works just inside your home or office network. If gadgets talk while hooked to one single switch, the box checks a list to send data right where it needs to go. This ID stays put until you swap out the gear. Think of it like a built-in tag for how stuff links up locally.

2. IP Address – Layer 3 (Network Layer)

A web address acts like a label to spot gadgets on various networks. Instead of staying fixed like a hardware ID, this number shifts when the gadget joins another network. One type uses 32 bits -

called version four; the other, bigger one uses 128 bits - known as version six.

Routers rely on IP addresses to figure out how data moves toward its target network. Although MAC addresses handle delivery within a local network, IPs manage travel across networks - guiding info straight from sender to receiver, no matter where they are online.

3. Port Number – Layer 4 (Transport Layer)

A port number points to a particular app or service on a machine. Even though an IP label shows the device, the port tag directs traffic to the right program at the transport stage.

For example:

- HTTP → Port 80
- HTTPS → Port 443
- DNS → Port 53
- FTP → Port 21

This mix of IP address along with port number helps various apps work on one gadget without messing up. Like, a machine can check websites while sending messages or playing videos nonstop since every task runs on its own port.

Part A Summary

In Part A, I looked into networking basics - like the OSI model, TCP/IP setup, what PDUs do, also various network addressing styles. Each layer handles its own job during comms; besides that, data moves across them in an organized way.