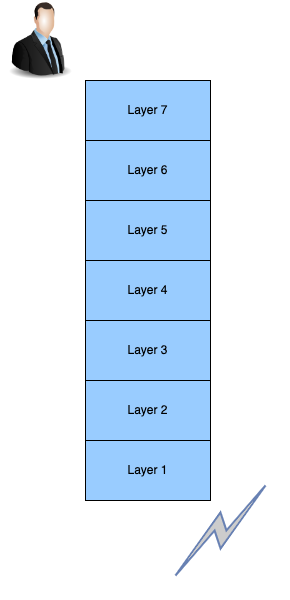
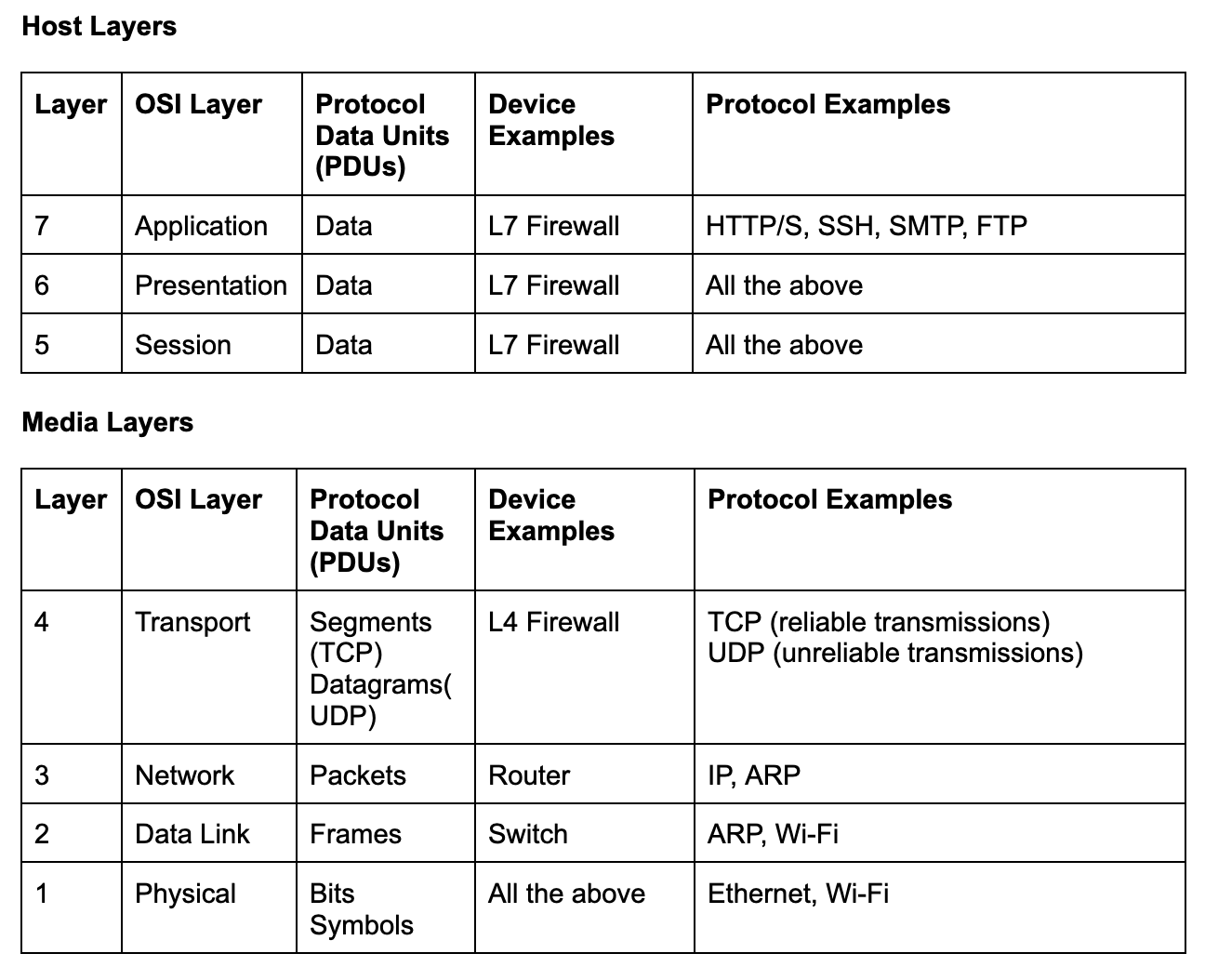
Network Reference Models

The OSI and TCP/IP Network Reference Models:  
These are conceptual models that provide abstract representations of how data is transmitted over a network. Each model is described by a “stack” of layers, so-called because each layer is dependent on its adjacent layers, and together they have an implied directionality.  


Data moving down the stack (sending data) undergoes encapsulation, where each layer of information is wrapped in additional details and passed to the next. This additional information typically takes the form of headers and/or trailers – details added to the beginning or end of the data units received from the previous layer. Once fully encapsulated and sent down the wire to the target system, that data is moved back up the stack, undergoing decapsulation, where each layer will interpret and remove (strip) this additional information until only the original data remains.  
  
The OSI stack has 7 such layers and the TCP/IP stack has 4. However, after you know one, you basically know the other. It’s not complicated, and it’s not arbitrary either

**The OSI Model**The Open Systems Interconnection model was created by ISO (International Standards Organization) in the 1970’s as a way to organize efforts around the creation of new networking protocols, encourage development of systems that can interoperate regardless of vendor, and to provide a common perspective for the existing network protocols at the time. In the early days of networking, most standards were either government-sponsored, or proprietary standards developed by vendors, which locked consumers into particular equipment and digital ecosystems. The OSI attempted to solve this with a universal description of the interactions between various network protocol implementations.

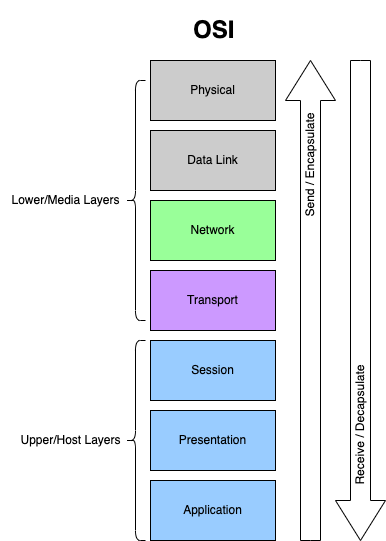
The OSI model has 7 layers. The bottom 4 are referred to as “media” layers, and comprise the most important constructs for addressing, data transport, and data delivery. The top 3 layers are referred to as “host” layers, and rely on the infrastructural scaffolding provided by the media layers to facilitate communication between applications.



Let’s quickly breakdown the function of each layer:

* **Layer 7 (Application)**- The human-computer interaction layer.  
    
  This layer handles the services and programs that use the network to transmit and receive data, such as web browsers and email clients.
* **Layer 6 (Presentation)**- Formatting and presentation of data.   
    
  This layer handles things like compression/decompression, encryption/decryption, and encoding/decoding. Basically, ensuring the data is in a usable format for the next layer (Layer 5 if sending data, Layer 7 if receiving it).
* **Layer 5 (Session)**- Managing communication sessions between devices.  
  This layer is responsible for setting up, maintaining, and tearing down sessions, in addition to performing functions like authentication and authorization. This ensures connections between clients are opened for as long as needed to transfer data, then torn down as soon as the transfer is complete.
* **Layer 4 (Transport)**- Transporting data between hosts.   
    
  This is the layer that handles ports. You know, like HTTP at port 80, HTTPS at port 443. Being responsible for data delivery, this layer also handles breaking large data transfers into pieces for delivery, and then reconstituting them at the other end.  
    
  There are two main protocols that are used at this layer: TCP and UDP.   
    
  TCP sends segments of data, and has delivery guaranteeing mechanisms in it, such as checking packets for errors, ensuring they are received in the correct order, and re-transmitting when packets are dropped. This ensures high fidelity communication, but these checks and balances come at a cost of speed. If you want speed – the name of the game when streaming media such as video games or movies – you use UDP.   
    
  UDP sends datagrams and doesn’t care whether they make it or not – fire them off, and yell “catch!” A few dropped packets is unlikely to ruin your kill streak, so you can do away with that error-checking overhead to satisfy your need for speed.
* **Layer 3 (Network)**- Transmitting data between devices in different networks, i.e., routing! This is the province of IP addresses, the logical addresses assigned to hosts.  
    
  Ever heard of a router? Routers are layer 3 devices that enable communication across networks by routing packets between those devices based on IP addresses.
* **Layer 2 (DataLink)**- Transmitting data between nodes on a network, i.e., switching, This is the province of MAC addresses, the physical addresses assigned to network cards.  
    
  Ever heard of a switch? Not the Nintendo one. Switches are Layer 2 devices that enable communication on a local network by forwarding frames between those devices. Ever heard of packets? Frames are like packets, but on Layer 2.
* **Layer 1 (Physical)**- Transmitting data over a physical medium, such as a cable or over-the-air.  
    
  The building blocks of communications between electronics – electrical signals and the various low-level data units they represent (bits, or binary digits, and symbols).

Presuming you’ve read all this, you deserve a picture. Here you go:



**Tip:**While primarily theoretical, you’ll often hear OSI layers in reference to devices in the networking world. For example, switches – those devices which connect computers on a local network – are typically Layer 2 devices. But, eventually switches were built with the ability to perform routing between various virtual networks (VLANs). Recall that routing is a Layer 3 activity that uses packets and is typically the province of routers, and so now it became necessary to distinguish between a Layer 2 and a Layer 3 switch. You’ll hear that differentiation often – thank us later.

Furthermore, firewalls saw a similar evolution. In their early years, they were simple packet filters, operating at Layer 3 (Network). Then came Layer 4 routers, which were able to keep track of states or connections established over TCP, an essential characteristic of the Transport Layer. Nowadays, we have next generation firewalls which can perform Layer 7 filtering, further reinforcing the necessity for clarifying the Layer with which the device is able to work.

Despite the OSI model’s residence primarily in theory, we promise you will not be able to help seeing this stack reflected in the networking products and protocols you use from hereon. Spare yourself the brain cycles in your future network explorations by locking this one down early.

Despite the OSI model not being widely adopted for practical implementations, it remains an important article in theoretical and academic discussions. In practice, it serves as a:

* Comprehensive theoretical framework for understanding network communications.
* Logical map of data flow that helps for isolating causes of network problems during troubleshooting.
* A common language for discussing network systems and devices that operate on specific layers of data.

**Mnemonic**There are many acronyms available for help memorizing these layers. Feel free to look for one you like, create your own, or adopt one of the examples provided below (one for each direction, depending on your preference):

*L7-L1*

*All People Seem To Need Data Processing*

*L1-L7*

*People Don’t Need Those Stupid Packets Anyway*

Alright, you’ve got the OSI down. Easy peasy – next!!

**The TCP/IP Model**If you felt like that was a lot of information and are reluctant to take on another model, do not fear: this one actually simplifies the stack by combining several of the layers you’re already familiar with.

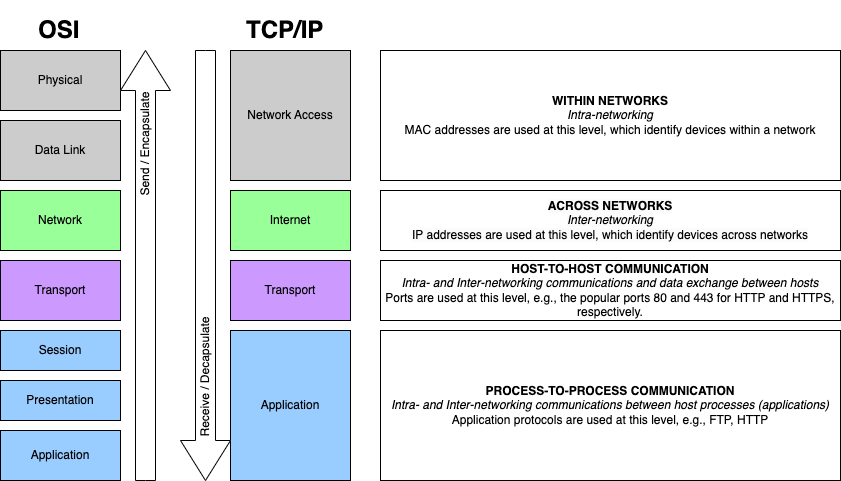
In contrast to the theoretical OSI model, the TCP/IP model is a practical model that maps more closely to communications in modern networks. Where the OSI failed to gain traction for practical implementations, the TCP/IP model won the province of modern network communications and was officially implemented as the Internet Protocol Suite, the backbone of the modern Internet. You know, like IP, that OSI Layer 3 (Network) protocol that handles routing between different networks (i.e., inter-networking)? Kind of a big deal.

Given your familiarity with the OSI model, it’s easy to recognize this one as a practically abridged version of it. The TCP/IP model combines OSI Layers 1-2 (Physical, Data Link) into a single Layer 1 (Network Access), and Layers 5-7 (Application, Presentation, Session) into a single Layer 4 (Application). It also renames the Network layer (4) to Internet, reflecting its titular protocol. This leaves you with the following:

* **Layer 1: Network Access**
* **Layer 2: Internet**
* **Layer 3: Transport**
* **Layer 4: Application**

Tip: Notice that the only layers in this model not collapsed from the OSI were the eponymous TCP and IP protocols (Transport and Network/Internet layers), those fundamental constructs of data delivery and routing. This provides an easy way to convert from OSI to TCP/IP – it’s right there in the name!

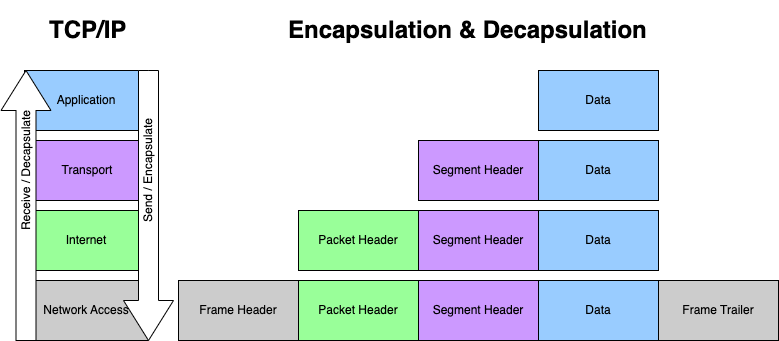
Since you’ve already slayed the great OSI dragon, you basically already have the TCP/IP model down pat. Here’s proof:



In practice, the TCP/IP model provides the following:

* Forms the basis of the Internet and other large-scale networks
* Provides a simplified, more practical approach to network communications, relative to the OSI model
* Serves as the universal standard for network communication, given its wide adoption and implementation by a wide variety of devices and applications.

Before we finish this section, we owe you one more picture. Remember the details around encapsulation (moving down the stack, or sending data) and decapsulation (moving up the stack, or receiving data)? You know, the process of adding to or removing data from the beginning or end of the data units passed by other layers? Here’s what that looks like, with reference to the TCP/IP model.



**Mnemonic**A TIN. That’s all – just imagine A TIN when you are attempting to recall the TCP/IP stack. Any kind of tin – just A TIN.

**In Conclusion**It is important to note that these models are not an either/or situation. Both are useful, and often in combination, rather than with any exclusivity (as you’ll see in the next section). The layers are important to understand, but more important is understanding the relationships between each. Once you have these relationships down, it’s easy to organize the communications you see as the province of particular functions, rather than numbered layers of any specific model – and THAT, dear C-Labber, is what will give you your networking wings. Both models are crucial for your understanding, and developing your technical linguistics.

Summarily, remember your mnemonics, look for these layers wherever you see network communications, know that they don’t exist in isolation, and, most importantly, USE them as a cheat sheet for understanding. We’re not foisting theory on you for the sake of it. These models are bred for making networking more understandable and universal, not for complicating the matter.