

Lesson Type : Knowledge/Information

Level : Intermediate

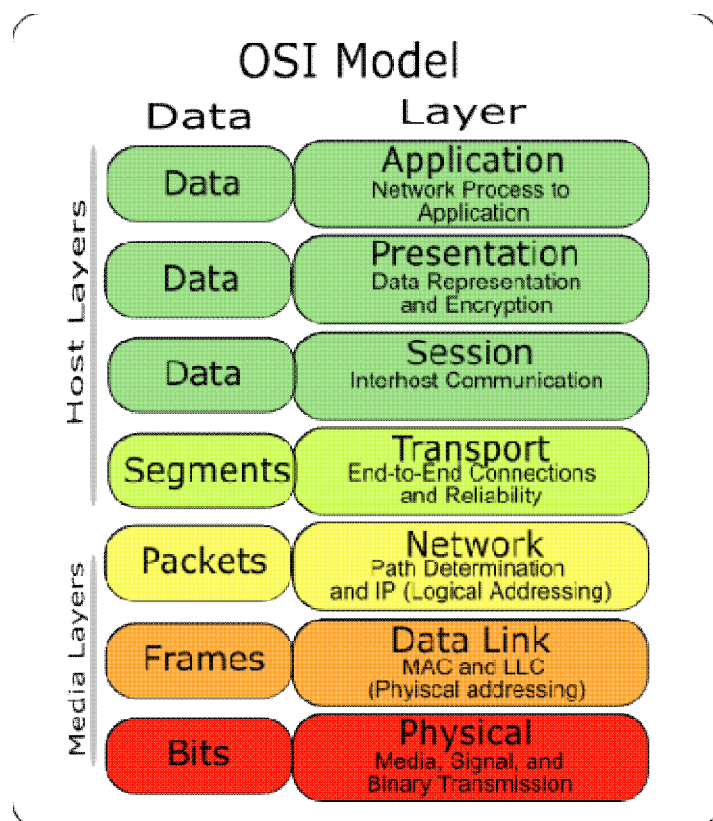
Scope : Network +

Characters : Round About 23700

Image : 2

Price : 200Ks

OSI (7) Layers အကြောင်းသိကောင်းစရာ



OSI ဆိုတာဟာဘာလဲ။ အရှည်ပြောရရင်တော့ Open System Interconnection လို့ခေါ်ပါတယ်။ သူဟာ Network တွေ တစ်ခုနဲ့တစ်ခု အပြန်အလှန် ချိတ်ဆက်ကြရာမှာ အခက်အခဲမရှိ လွယ်ကူချောမွေ့ စေဖို့အတွက် International Organization for Standardization (ISO) အဖွဲ့ ကြီးက ချမှတ်ထားတဲ့ စံတစ်ခုဖြစ်ပါတယ်။

ကမ္ဘာပေါ်မှာ ကွန်ပျူတာ ထုတ်လုပ်ရောင်းချတဲ့ ကုမ္ပဏီကြီးတွေ အများကြီးရှိပါတယ်။ အဲဒီလိုပဲ ကွန်ပျူတာတွေ တစ်လုံးနဲ့ တစ်လုံးချိတ်ဆက်တဲ့အခါမှာသုံးတဲ့ Networking Devices တွေကို ထုတ်လုပ်တဲ့ ကုမ္ပဏီတွေလည်း အများကြီးရှိပါတယ်။ အဲဒီလိုမျိုး နည်းပညာပိုင်းဆိုင်ရာ မတူညီကြတဲ့ အမျိုးအစားအမျိုးမျိုးသော ကွန်ပျူတာတွေ၊ Networking Devices တွေကို အသုံးပြုထားကြတဲ့ Network တွေအသီးသီးဟာ တစ်ခုနဲ့တစ်ခု လွယ်လင့်တကူ အပြန်အလှန် ချိတ်ဆက်မိချင်တယ်ဆိုရင် သူတို့အားလုံးကြားထဲမှာ ပုံသေစံနှုန်း(ဘုံတူညီချက်များ) ရှိဖို့တော့လိုအပ်နေပါတယ်။ ဒါမှသာ အဲဒီ Network တွေဟာ အတူတကွ ချောမွေ့စွာ ချိတ်ဆက် အလုပ်လုပ်နိုင်ကြမှာဖြစ်ပါတယ်။ဒါကြောင့် ISO အဖွဲ့ကြီးဟာ အဲဒီလိုအပ်ချက်ကို ပြေလည်စေဖို့အတွက် OSI Layers (7)ခုကို ၁၉၇၄ ခုနှစ်လောက်ကတည်းက စတင်ဖော်ထုတ်ခဲ့တာပဲဖြစ်ပါတယ်။

အခုအချိန်မှာတော့ OSI(7) Layers စံနှုန်းများနဲ့ တိုက်ဆိုင်ကိုက်ညီမှုမရှိဘဲ ကွဲပြားနေတဲ့ Networking နဲ့ပတ်သက် ဆက်သွယ်နေသော Device ရယ်လို့ ဈေးကွက်ထဲမှာ မရှိတော့တဲ့အထိ တစ်ကမ္ဘာလုံးက OSI Layers (7) ခုကို အသိအမှတ်ပြုလက်ခံ နေကြပါပြီ။ ဘာဖြစ်လို့လဲဆိုတော့ လူသားတွေဟာ ယနေ့ခေတ်မှာ ဆက်သွယ်ရေးစနစ်ရဲ့အရေးကြီးပုံနဲ့အသုံးဝင်ပုံ တွေကိုပိုမိုခံစား သိရှိလာကြတာနဲ့အမျှ ကောင်းမွန်မြန်ဆန်တဲ့ ဆက်သွယ်ရေးစနစ်တွေကို အသုံးပြုခွင့်ရဖို့ ပိုမိုတောင့်တလာကြပါတယ်။ ကွန်ပျူတာ ကွန်ယက်တွေဟာလည်း ယနေ့ဆက်သွယ်ရေးစနစ်တွေမှာ အရေးပါတဲ့ အစိတ်အပိုင်းတွေအဖြစ် ပါဝင်လှုပ်ရှားနေကြတယ်ဆိုတာ အားလုံးသိနေကြမယ်ထင်ပါတယ်။ ဒီတော့ ကွန်ပျူတာ ကွန်ယက်တွေကြားမှာ နည်းပညာ Platform မတူသည်ဖြစ်စေ၊ အမျိုးအစား တံဆိပ်မတူသည်ဖြစ်စေ အတူတကွ ချိတ်ဆက်လုပ်ဆောင် နိုင်ကြဖို့ဆိုတာ ဟာလည်း အမှန်တကယ်အရေးကြီးတဲ့လိုအပ်ချက် တစ်ခု ဖြစ်ပါတယ်။ ဒါမှသာ ကျယ်ပြန့်ကြီးမားတဲ့ ကွန်ပျူတာကွန်ယက်တွေ စီမံဖန်တီးချိတ်ဆက်နိုင်မှာပါ။ အဲဒီလိုအဲဒီလို အချက်တွေကြောင့်

လည်း OSI(7) Layers ဟာ ကမ္ဘာ့နည်းပညာစင်မြင့်ထက်မှာ အများစုလက်ခံသုံးစွဲတဲ့ စံနှုန်းတစ်ခုအဖြစ်အလျင်အမြန် နေရာရလာခဲ့ တာပဲဖြစ်ပါတယ်။ OSI(7) Layers စံနှုန်းပေါ်ပေါက်လာပြီးတဲ့နောက်ပိုင်း ကွန်ပျူတာကွန်ယက်ပိုင်းဆိုင်ရာ နည်းပညာတွေဟာ အပြန်အ လှန် ပူးပေါင်းဆောင်ရွက်မှုတွေနဲ့အတူ အလျင်အမြန်တိုးတက်လာခဲ့ပါတယ်။ ကွန်ပျူတာကွန်ယက်တွေရဲ့ လွှမ်းခြုံနှံ့ကျက်မှုဧရိယာ တွေဟာလည်း ပိုမိုကျယ်ပြန့်ကြီးမားလာခဲ့ပါတယ်။ အသိသာအမြင်သာဆုံး အကျိုးဆက်ကို ပြပါဆိုရင်တော့ အခုတစ်ကမ္ဘာလုံး တွင်တွင်ကျယ်ကျယ် အသုံးပြုနေကြတဲ့ Internet ကိုသာကြည့်ပါ။ Internet ဆိုတာကတော့အားလုံးသိနေကြတဲ့ အတိုင်း ကမ္ဘာပေါ် ကလူသန်းပေါင်းများစွာတို့ရဲ့ ဆက်သွယ်ရေးကြားခံမီဒီယာတစ်ခု လို့ပြောရင်လဲရသလို လိုအပ်တဲ့ သတင်းအချက်အလက် အများဆုံးကိုအမြန်ဆုံးရှာဖွေ ရယူနိုင်တဲ့တစ်ကမ္ဘာလုံးဆိုင်ရာ ကွန်ပျူတာကွန်ယက်ကြီးပဲဖြစ်ပါတယ်။ Internet ပေါ်မှာရှိတဲ့ ကွန်ပျူတာတွေ၊ Networking Devices တွေအားလုံးဟာ OSI(7) Layers စံနှုန်းတွေနဲ့အညီ အချင်းချင်း ချိတ်ဆက်လုပ်ဆောင်နေကြ တာပဲဖြစ်ပါတယ်။ သူတို့တွေထဲမှာ မတူညီတဲ့တံဆိပ်အမျိုးအစားတွေအများကြီး ပါတယ်။ မတူညီတဲ့ နည်းပညာ Platform တွေအများကြီးလည်းပါပါတယ်။ အားလုံးဟာ အတူတကွ ပြဿနာမရှိဘဲ အချင်းချင်း ချိတ်ဆက်လုပ်ဆောင်နေကြတာဖြစ်ပါတယ်။ ကဲ ဒီလောက်ဆိုရင်တော့ OSI(7) Layers ဆိုတာဘာလဲဆိုတာ ရယ် ဘယ်လိုဘယ်ပုံပေါ်ပေါက်လာတယ်ဆိုတာတွေကိုသိမြင်လောက်ပြီ ထင်ပါတယ်။

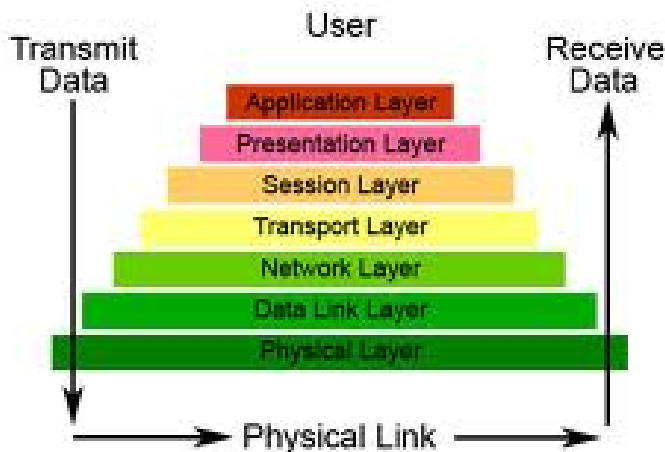
OSI(7) Layers ရဲ့ပေါ်ပေါက်လာပုံ နဲ့ ဘာကြောင့် ပေါ်ပေါက်လာရတယ်ဆိုတာတွေကို သိရပြီးမှတော့ အဲဒီ Layers (၇) ခုအကြောင်းကိုကော ပိုပြီးမသိချင်ကြဘူးလား။ သိချင်တယ်ဆိုရင်တော့ ကျွန်တော် တတ်နိုင်သလောက်ကြိုးစားပမ်းစား ရှင်းပြ ထားတာလေးတွေကို ဆက်ပြီးဖတ်လိုက်ပါဦးနော်။

အရင်ဆုံး Layers (၇) ခု ရဲ့နာမည်လေးတွေကို အရင်ပြောပြမယ်နော်ဟုတ်ပြီလား။ သေသေချာချာလေးမှတ်သားထားမယ် ဆိုရင်တော့ပိုကောင်းတာပေါ့နော်။ Networking ကိုစိတ်ဝင်စားတယ်ဆိုတဲ့ လူတစ်ယောက်အနေနဲ့ ဒီ OSI Layers (၇) ခုဟာ ဘာတွေလဲဆိုတာလောက်တော့အနည်းဆုံး သိထားသင့်တယ်ဗျ။ ကဲကဲ မှတ်ထားလိုက်ကြနော်။

- (၇) Application Layer
- (၆) Presentation Layer
- (၅) Session Layer
- (၄) Transport Layer
- (၃) Network Layer
- (၂) Data Link Layer နဲ့
- (၁) Physical Layer တို့ပဲဖြစ်ပါတယ်။

အဲ ကျွန်တော်က User နဲ့အနီးစပ်ဆုံး Layer ကနေစပြီး ရေးလိုက်တော့ Application Layer ကထိပ်ဆုံး ရောက်သွားတယ်။ ဒါပေမယ့် Network အခေါ်အဝေါ် အရဆိုရင်တော့ Application Layer ဟာ Layer (၇)ပါ။ အဲဒါလေးကိုလဲမရောထွေးစေဘဲ သေသေချာချာ မှတ်သားထားဖို့ သတိပေးလိုက်ပါရစေနော်။ နောင်မှာ Layer နံပါတ်နဲ့ ပြောကြရေးကြတာတွေကို တွေ့တဲ့အခါ ရှုပ်ထွေးကုန်မှာ စိုးလို့ပါ။ ကျွန်တော်ကတော့ User နဲ့အနီးစပ်ဆုံး Layer ကနေပဲအစဉ်လိုက် တစ်ခုစီ ရှင်းပြသွားပါမယ်။ သေသေချာချာလေး ခေါင်းကိုအေးအေးဆေးဆေးထားပြီး တွေးလိုက်ငေးလိုက်(အဲလေ) တွေးလိုက်ဖတ် လိုက် နဲ့တွေးတွေးပြီး ဖတ်ပြီးသမျှကို နားလည်အောင် စဉ်းစားပါနော်။ နားလည်သွားရင်လည်း တစ်ခါတည်းသာ မှတ်သားထားလိုက် ပါနော်။ Networking နဲ့ပတ်သက်တဲ့ ပစ္စည်းကိရိယာအားလုံးဟာ OSI Layers(၇)ခု နဲ့ကင်းကွာလို့မရနိုင်ပါဘူး။ ဒီ OSI Layers(၇)ခု ပေါ်မှာပဲအခြေခံပြီး အလုပ်လုပ်ကြတာမို့ Network သမားတစ်ယောက် အနေနဲ့ ဒီ OSI Layers(၇)ခု အကြောင်းကို အနည်းဆုံး အကြမ်းဖျင်းလောက်တော့နားလည် ထားသင့်ပါတယ်။

## The Seven Layers of OSI



အရင်ဆုံး Layer(၇)ခုတစ်ခုနဲ့တစ်ခု ဆက်သွယ်ပြီး အလုပ်လုပ်ပုံကို အပေါ်ယံမြင်နိုင်အောင် ပုံဆွဲပြလိုက်ပါတယ်။ ပုံထဲမှာ Computer A ကနေ Computer B ကို Data Packet သို့မဟုတ် Message တစ်ခုပို့လိုက်တယ်ပေါ့ဗျာ။ အဲဒီ Computer A ကပို့လိုက် တဲ့ Message ကို Computer B ဆီကိုတန်းတန်းမတ်မတ် စိတ်ချရတဲ့အနေအထား (Message အစစ်အမှန်အပြည့်အစုံ) နဲ့ ရောက်ရှိ သွားနိုင်ဖို့အတွက် OSI Layers(၇)ခုလုံးဟာ တစ်ခုနဲ့တစ်ခု အဆင့်ဆင့် လက်ဆင့်ကမ်း ပူးတွဲဆောင်ရွက်သွားကြတာပါ။ အဲဒီလို တစ်ခုနဲ့တစ်ခု အဆင့်ဆင့် ပူးတွဲဆောင်ရွက်ကြပုံကြောင့် OSI(7) Layers ကို တစ်ခါတစ်ရံမှာ Protocol Stacks လို့လဲ ခေါ်ကြပါတယ်။

ကဲကဲ အပေါ်ယံ လေ့လာကြည့်ပြီးရင်တော့ အသေးစိတ်ဆက်ပြီး လေ့လာကြတာပေါ့။ ပထမဦးဆုံးအနေနဲ့ လေ့လာရမှာက တော့ Application Layer(Layer 7) ပါပဲ။ Application Layer ကဘာတွေလုပ်သလဲ။ပြောရရင်တော့ သူက Layer(၇)ခုလုံးမှာ User နဲ့ အနီးစပ်ဆုံး Layer ပါပဲ။ User တစ်ယောက် Network Connection အသုံးပြုပြီး Data တွေကိုဟိုဘက်တခြား User တစ်ယောက်ဆီကို ပို့ဖို့ ဒါမှမဟုတ် ဟိုဘက်ကပို့လိုက်တဲ့ Data တွေကို ပြန်လည်မြင်တွေ့ အသုံးပြုနိုင်အောင်လို့ လိုအပ်တဲ့ Services တွေ Application တွေ အားလုံးဟာ ဒီ Layer မှာ အကျုံးဝင်ကြပါတယ်။ ဥပမာ Electronic Mail တို့၊ File Transfer တို့၊ Remote Job Entry တို့ စတဲ့ Service တွေပေါ့။ ပြောရရင် Application Layer ဟာ User တွေကို သူတို့ ပို့ချင်တဲ့ Data အမျိုးအစားတွေအလိုက် သူတို့ပို့ချင်တဲ့ ပုံစံအနေအထားအတိုင်း လွယ်လွယ်ကူကူနဲ့ Network Connection ပေါ် တင်ပေးနိုင်အောင် ကူညီပံ့ပိုးပေးပါတယ်။ Network Connection ကိုဖြတ်သန်းပြီး

လက်ခံမယ့်သူဘက်အခြမ်းကို ရောက်သွားရင်လည်း ဒီဘက်ကပို့လိုက်တဲ့ Data အမျိုးအစားကို လိုက်ပြီး လက်ခံသူအဆင်ပြေစွာနဲ့ အဲဒီ Data တွေကိုလက်ခံယူနိုင်အောင်၊ ပြန်လည်အသုံးပြုနိုင်အောင် လိုအပ်တဲ့ Service တွေနဲ့ကူညီပေးပြန် ပါတယ်။ ဘယ်လိုသဘောမျိုးလဲဆိုတော့ Interface တစ်ခုလိုသဘောထားမယ်ဆိုရင်လည်းရပါတယ်။ ကျွန်တော်တို့အနေနဲ့ကွန်ပျူတာ ထဲမှာ စာစီစာရိုက်လုပ်ချင်တယ်။ ဒါမှမဟုတ်ကျွန်တော်တို့ရေးသားထားတဲ့စာတွေကို ကွန်ပျူတာထဲမှာသိမ်းထားချင်တယ်ဆိုရင် Microsoft Word တို့၊ Adobe PageMaker တို့ဟာ ကွန်ပျူတာနဲ့ ကျွန်တော်တို့ကြားကနေ ကျွန်တော်တို့ကို ကူညီပေးတဲ့ Interface တွေပဲဖြစ်ပါတယ်။ အဲဒီလိုပဲ ကွန်ပျူတာတွေ တစ်လုံးနဲ့တစ်လုံး ချိတ်ဆက်ပြီး သတင်းအချက်အလက်တွေလှယ်တဲ့အခါ ဟိုဘက်ဒီဘက် Data တွေ Transfer လုပ်ကြတဲ့အခါ တွေမှာလည်း User တွေနဲ့ Network Connection တွေကြား ပြောရရင်တော့ User နဲ့ အောက်ပိုင်း Layer တွေကြားမှာ Application Layer ဟာ Interface တစ်ခုပါပဲ။ User တွေပို့ချင်တဲ့ သတင်းအချက်အလက်တွေ၊ Data တွေအားလုံးကို အောက်ပိုင်း Layers တွေကနားလည်လက်ခံယူနိုင်ပြီး ဟိုဘက်တခြားကို ဆက်လက်ပို့ဆောင်ပေးနိုင်ဖို့အတွက် Application Layer ကကြားကနေပင်ရောက်ပြီး Interface တစ်ခုအနေနဲ့ ဆောင်ရွက်ပေးတာဖြစ်ပါတယ်။ ဒီလိုဆောင်ရွက်ပေးရာမှာ User ပို့လွှတ်မယ့် Data အမျိုးအစားပေါ်မူတည်ပြီး အသုံးပြုတဲ့ Service တွေဟာလည်း အမျိုးမျိုးရှိနေမှာဖြစ်ပါတယ်။ ဒါမှလည်း Data အမျိုးအစားများစွာကို User တွေအခက်အခဲမရှိဘဲ တစ်ဖက်နဲ့တစ်ဖက် Transfer လုပ်နိုင်မှာဖြစ်ပါတယ်။

ဥပမာတစ်ခုနဲ့ပိုရှင်းအောင် ပြောရမယ်ဆိုရင်တော့ ကျွန်တော်တို့ တယ်လီဖုန်းဆက်သွယ်ရေးစနစ်နဲ့ ဥပမာပေးချင်ပါတယ်။ အဲဒီမှာ တယ်လီဖုန်းအခွံ (Hand Phone မှာဆိုရင်တော့ HandSet ပေါ့နော်) ဟာ Application Layer လို့သဘောထားပေးပါ။ အဲဒီဖုန်း ရဲ့နံပါတ်ကတော့ Application Layer ကအသုံးပြုတဲ့ Service ပေါ့နော်။ Data Transfer လုပ်ချင်တယ်ဆိုရင် Network Connection ရှိရမှာဖြစ်သလို အဲဒီဖုန်းဟာလည်း Service ပင်နေဖို့တော့လိုတာပေါ့နော်။ ကဲဒါဆိုရင် ကျွန်တော်တို့ အဲဒီဖုန်းကို အသုံးပြုပြီးတော့ ကျွန်တော်တို့ရဲ့သူငယ်ချင်းတွေ မိတ်ဆွေတွေဆီကို ဆက်သွယ်လို့ ရနိုင်ပြီပေါ့။ အဲဒီမှာ ကျွန်တော်ပြောချင်တာက ကျွန်တော်တို့ ခေါ်ဆိုချင်တဲ့ Phone ရဲ့အမျိုးအစား(GSM လား၊ WCDMA လား၊ CDMA 450 လား ဒါမှမဟုတ် ကြိုးဖုန်းလား) ဆိုတာပေါ်မူတည်ပြီး ကျွန်တော်တို့အသုံးပြုခေါ်ဆိုရတဲ့ ဖုန်းနံပါတ်တွေဟာမတူကြဘူးမဟုတ်လား။ အင်းအဲဒီလိုပါပဲ Application Layer ဟာလည်း ကျွန်တော်တို့အသုံးပြုချင်တဲ့ Data အမျိုးအစားအလိုက် Services တွေအများကြီးတွေနဲ့ ကြားခံ Interface တစ်ခုအဖြစ်ကျွန်တော်တို့နဲ့ အောက်ပိုင်း Layer တွေ ကြားမှာ ကူညီလုပ်ဆောင်ပေးပါတယ်။

ကဲကဲ Application Layer တစ်ခုတည်းကိုရှင်းနေတာတော်တော်တောင်ရှည်နေပြီ။ ဒီလောက်ဆိုရင်လည်း Application Layer အကြောင်းကို တော်တော်လေးသဘောပေါက်လောက်ပြီလို့ ယူဆပါတယ်။ ဒါဆိုရင်အောက်ကနောက်ထပ် Layer တစ်ခုကို ထပ်ပြီးလေ့လာကြတာပေါ့နော်။ နောက်ထပ် Layer တစ်ခုကဘာပါလိမ့်။ မှတ်မိတဲ့လူရှိရင်လည်း ပြောလိုက်ကြပါနော်။ ဘာတဲ့။ ဟုတ်ကဲ့။ Presentation Layer ပါ။ မှန်ပါတယ်။ အားလုံးမှန်ကြပါတယ်။ ကဲကဲဆက်ပြီး လေ့လာလိုက်ကြရအောင်နော်။

Presentation Layer(Layer 6) ကတော့ Application Layer ကနေ စီးဆင်းလာတဲ့ Data တွေကို Network Connection တစ်လျှောက်မှာ သွားလာနိုင်မယ့် Format ပုံစံမျိုးကို ပြောင်းလဲပေးပါတယ်။ အဲဒါကို Encode လုပ်တယ်လို့ခေါ်ပါတယ်။ ဒါ့အပြင် Security ပိုင်းစိတ်ချရစေရန်အတွက်လိုအပ်တဲ့ Encryption တွေကိုလည်းလုပ်ပေးပါတယ်။ ပြီးတော့ပမာဏကြီးမားစွာဖောင်းပွနေတာ မျိုးမဖြစ်အောင် လိုအပ်တဲ့ Compression လုပ်ဆောင်မှုအပိုင်းလေးတွေလည်း လုပ်ဆောင်ပေးပါတယ်။ ဒါကပို့တဲ့ဘက်ကပေါ့နော်။ လက်ခံမယ့်ဘက်ရောက်တဲ့အခါကျတော့ ပုံမှန်အရွယ်ပမာဏပြန်ဖြစ်အောင် မှန်မှန်ကန်ကန် Decompression လုပ်ပေးပါတယ်။ မှန်ကန်သော Encryption နည်းလမ်းများနဲ့ Decryption ပြန်လုပ်ပေးပါတယ်။ ပြီးရင်တော့ သင့်တော်တဲ့ Format တစ်ခုနဲ့ Data တွေ ကိုပြန်ပြီး Decode လုပ်ပေးပါတယ်။ Application Layer ကနားလည်လက်ခံနိုင်မယ့် Format မျိုးနဲ့ပေါ့။ ဒါမှလည်း Application Layer က User တွေကို Data တွေပြန်လည်ပြသပေးနိုင်မှာဖြစ်ပါတယ်။

အင်းအဲဒီလိုလုပ်ဆောင်တဲ့နေရာမှာ Application Layer မှာတုန်းကလုပ်ခဲ့နည်းစနစ်အမျိုးမျိုးရှိပါတယ်။ Encodeလုပ်တဲ့နေရာ မှာနဲ့ Decode လုပ်တဲ့နေရာမှာဆို သုံးရတဲ့ Mechanism တွေက ပိုတဲ့ဘက်အခြမ်းမှာ နဲ့လက်ခံမယ့်ဘက်အခြမ်းမှာတောင် တူချင်မှတူ ပါလိမ့်မယ်။ဘာလို့လဲဆိုတော့ ဒီဘက် Computer A(Sender) ကသုံးတဲ့ Character Set(ကွန်ပျူတာကသုံးတဲ့ Data Representation စနစ်) နဲ့ဟိုဘက်က Computer B(Receiver) ကသုံးတဲ့ Character Set နဲ့က တူချင်မှတူမှာမို့ပါ။ဥပမာ Computer A က ASCII Code (American Standard Code for Information Interchange) ကိုအသုံးပြုပြီး သူ့ရဲ့စက်မှာ Data တွေကိုသိမ်းပေးမယ့် Computer B ကတော့ EBCDIC (Extended Binary Coded Decimal Interchange Code) ကိုသုံးချင်သုံးပါလိမ့်မယ်။ဒါဆိုရင် Presentation Layer ဟာပိုတဲ့ဘက် Sender ဘက်မှာ ASCII Coded Data တွေကို Network Connection ပေါ်မှာသွားလာမယ့် Format မျိုးအဖြစ်ပြောင်း လဲပေးရပြီးလက်ခံမယ့်ဘက်မှာတော့ Network Connection ပေါ်ကလာတဲ့ Data တွေကို EBCDIC Coded Data တွေအဖြစ်ပြန်လည် ပြောင်းလဲပေးရပါတယ်။ ဒါဟာ Presentation Layer ရဲ့အဓိကလုပ်ဆောင်ချက်ပါ။

နောက်ပိုင်း Network Connection တွေပေါ်ကနေ Data တွေကို အပြန်အလှန် Transfer လုပ်ကြရာမှာ Security ပိုင်းဟာ လည်းစဉ်းစားစရာ အချက်တစ်ချက် ဖြစ်လာခဲ့ပါတယ်။ ဒါကြောင့် Data တွေကို Transfer လုပ်တဲ့နေရာမှာ ဟိုးအရင်လိုဒီအတိုင်းပုံမှန် အတိုင်း ပေးပို့ရာကနေ Encryption လုပ်ပြီးမှသာ Transfer လုပ်လာကြပါတယ်။ အဲဒီလို Encryption လုပ်တဲ့နေရာမှာတော့ဟိုဘက် နဲ့ဒီဘက်နှစ်ဘက်စလုံးမှာတူညီတဲ့လိုက်ဖက်ညီတဲ့စနစ်တစ်ခုကိုပဲသုံးကြပါတယ်။ Encoding နဲ့ Decoding တို့မှာလိုဟိုဘက်ဒီဘက်ကွဲ ပြားတဲ့ Mechanism မရှိတော့ဘူးလို့ ဆိုချင်တာပါ။အဲဒီလို Encryption Decryption လုပ်ဆောင်ချက်တွေဟာလည်း Presentation Layer မှာပဲလုပ်ဆောင်ကြတာပါ။ ပြီးတော့ Network Connection ပေါ်ကနေ Data တွေ Transfer လုပ်ကြရာမှာ ကျွန်တော်တို့အနေ နဲ့ Connection Bandwidth ကိုလဲမမေ့ထားသင့်ပါဘူး။ ဒီတော့ Data တွေ ဟိုဘက်ပို့ချင်တဲ့နေရာဆီကို မြန်မြန်ရောက်သွားဖို့ဆိုတာ ဟာ Data ရဲ့Size ပမာဏပေါ်မူတည်ပါတယ်။ Size သေးလေပိုမြန်လေပဲပေါ့။ ဒီတော့ Data တွေကိုပုံမှန် Size အတိုင်းပို့မယ့်အစား Compression လုပ်ပြီးမှပဲ ပို့ကြပါတယ်။အဲဒီလို Compression လုပ်ထားတဲ့ Data ကိုဟိုဘက် Destination (Computer B)ဆီရောက် တဲ့အခါ Decompression လုပ်ပြီးပုံမှန် Size ပြန်ရအောင်လုပ်ပါတယ်။ ဒီနေရာမှာတော့ Compression လုပ်တဲ့နည်းစနစ်နဲ့ Decompressionလုပ်တဲ့နည်းစနစ်တို့တူညီမှသာမှန်ကန်တဲ့ Data ကို မှန်ကန်တဲ့ Original Size အနေအထားပြန်ရမှာပါ။ အဲဒီလို Data Compression တွေ Decompression တွေအားလုံးဟာလည်း Presentation Layer မှာလုပ်ဆောင်ကြပါတယ်။

Session Layer(Layer 5) ကတော့ Data တွေ Transfer လုပ်ဖို့အတွက်ဟိုဘက်နဲ့ဒီဘက်ကြားထဲမှာ Connection တစ်ခုကို အစပြုဖြစ်ထွန်းစေပါတယ်။ Logical Connection ပေါ့နော်။ပြီးရင်အဲဒီ Connection လေးကို Data တွေ Transferလုပ်တာမပြီးသေးခင် အထိ ထိန်းသိမ်းထားပါတယ်။ Presentation Layer ကနေတဆင့်လက်ခံရရှိလာတဲ့ Data တွေကိုလည်းအဲဒီ Connection ပေါ်မှာ အဆင်ပြေပြေသွားနိုင်အောင်သင့်တော်တဲ့အပိုင်းလေးတွေအဖြစ်ပိုင်းပေးတာကိုလည်းလုပ်ဆောင်ပါသေးတယ်။အဲဒီလိုအပိုင်း လေးတွေ ပိုင်းတဲ့နေရာမှာအပိုင်းလေးတွေရဲ့ Size ပမာဏဟာရရှိနိုင်မယ့် Network Connection ပေါ်မှာမူတည်ပါတယ်။ အပေါ်မှာပြောခဲ့တဲ့ Logical Connection ဆိုတာလေး။ ဒါပေမယ့် Connection ရဲ့ Speed နဲ့ Bandwidth ပေါ်မှာမူတည်တာတော့မဟုတ်ပါဘူး။ အဲဒီ Connection ရဲ့အသုံးပြုထားတဲ့နည်းပညာပေါ်မှာသာမူတည်တာပါ။ အပိုင်းတစ်ပိုင်းမှာ ဘယ်လောက် Bytes ရှိမယ်ဆိုတာဟာအဲဒီ Connection တစ်လျှောက်လုံးကကြားခံ မီဒီယာကြိုးတွေ၊ Network Card တွေ၊ Switch တွေ၊ Router တွေ နဲ့ Protocols တွေပေါ်မှာ သာမူတည်မှာပါ။ ဆိုလိုတာက Ethernet လား၊ ISDN လား၊ ATM လား ဆိုတာတွေပေါ့။ ဒါတွေအားလုံးကို Session Layer ကလေ့လာ မှတ်သားပြီး Computer A နဲ့ Computer B ကြားမှာ Logical Connection လေးတစ်ခုကို စတင်ချိတ်ဆက်လိုက်ပါတယ်။ အဲဒီ Connection တစ်လျှောက်လုံးမှာဘယ်လောက် Size ပမာဏရှိတဲ့ Data Packet တွေသွားရင်အဆင်ပြေမယ်ဆိုတာကို သတ်မှတ်ပြီး Decision ချပါတယ်။ အဲဒီ Decision နဲ့အညီ Data တွေကို အပိုင်းလေးတွေပိုင်းပါတယ်။ အဲဒီလို ပိုင်းတဲ့အခါအပိုင်းလေးတွေတိုင်းမှာ လိုအပ်တဲ့ Information လေးတွေကိုလည်း တခါတည်းပေါင်းထည့်ပေးပါတယ်။

ဒီအပိုင်းလေးဟာ နံပါတ်ဘယ်လောက်ဖြစ်ပါတယ် ဆိုတာမျိုးပေါ့။ ဒါမှလည်းဟိုဘက်လက်ခံမယ့် Receiver ဘက်မှာ ပြန်ပြီးစုစည်းတဲ့အခါအစီအစဉ်အလိုက်မှန်မှန်ကန်ကန် စုစည်းနိုင် တော့မှာပေါ့။ အဲဒီလိုပိုင်းထားတဲ့ Data အပိုင်းလေးတွေကို အောက်ပိုင်း Layer တွေဆီကို ပို့ဆောင်လက်ဆင့်ကမ်းပေးပြီး ဟိုဘက်လို ချင်တဲ့ နေရာဆီကို ရောက်အောင်ဆက်လက်ပို့ဆောင်စေပါတယ်။ ဟိုဘက် Session Layer ကလည်းလက်ခံရရှိလာတဲ့ Data အပိုင်း လေးတွေကို နဂို Data ပြန်ရအောင်အစီအစဉ်အလိုက်ပြန်လည်ထုပ်ပိုးပါတယ်။ နဂိုမူရင်း Data ကိုရရှိပြီဆိုတော့မှ Presentation Layer ကိုပြန်လည်ပို့ဆောင်ပေးပါတယ်။ အဲဒီလို Data ကိုအပိုင်းလေးတွေပိုင်းတာ၊ ပြန်လည်စုစည်းတာဟာ Session Layer ရဲ့အဓိက လုပ်ဆောင်မှုဖြစ်ပါတယ်။ Logical Connection တစ်ခုအစပြုတည်ဆောက်တာဟာလည်းအဲဒီလို Data တွေပိုင်းတဲ့လုပ်ဆောင်မှုကို ထောက်ပံ့မှုရစေဖို့အတွက်ပဲဖြစ်ပါတယ်။ Logical Connection တစ်ခုတည်ဆောက်လိုက်ခြင်းဖြင့် Session Layer ဟာ တကယ့် Network Connection ပေါ်မှာ Data Transfer အဆင်ပြေပြေလုပ်လို့ရနိုင်မယ့် အခြေအနေတစ်ခုကို တိတိကျကျ သိရှိရအောင် စုံစမ်း လိုက်ခြင်းပဲဖြစ်ပါတယ်။ ဒါမှလည်း Data တွေကို ပိုင်းတဲ့အခါမှာ Network Connection ပေါ်မှာ ရနိုင်မယ့်အခြေအနေ၊ Services တွေ နဲ့ အဆင်ပြေအောင်ကောင်းမွန်စွာ ပိုင်းထုတ်နိုင်မှာဖြစ်ပါတယ်။ ကဲဒီလောက်ဆိုရင်တော့ Session Layer အကြောင်းကို တွေးမြင် သဘောပေါက်လောက်ပြီလို့ထင်ပါတယ်။

Transport	Layer(Layer	4)	ကတော့ကျွန်တော်တို့	Data
				Transfer

တွေအပြန်အလှန်ဖလှယ်ကြရာမှာစိတ်ချရတဲ့အနေအထားမျိုးနဲ့ လုပ်နိုင်အောင်လုပ်ဆောင်ပေးပါတယ်။ ဘယ်လိုလုပ်ဆောင်ပေးလဲဆိုတော့ခုနက Session Layer ကနေအပိုင်းလေးတွေ ပိုင်းပြီးလက်ဆင့်ကမ်းပေးလိုက်တဲ့ Data အပိုင်းလေးတွေကို (Data Packet လေးတွေပေါ့ဗျာ) သူက Additional Information လေး တွေထပ်ထည့်ပေးလိုက်ပါတယ်။ ဘာ Information လေးတွေလဲဆိုတော့ ပိုပြီးတိကျတဲ့ Packet Information လေးတွေ ဆိုလိုတာက ဒီ Packet မှာ Data ကဘယ်လောက် Bytes ပါတယ်၊ တခြား Information တွေလည်း ဘယ် Information ကဖြင့်ဘယ်လောက် Bytes ပါတယ် စသဖြင့်ပေါ့ဗျာ။ အဲဒီလို Information လေးတွေအပြင် Packet တိုင်းမှာ Error Control လေးတွေလည်းထပ်ထည့်ပေး သေးတယ်ဗျာ။ အဲဒီ Error Control လေးတွေကတော့ နည်းစနစ်အမျိုးမျိုးနဲ့ အများကြီးရှိပါတယ်။ သူတို့အားလုံးကဒီ Packet ကိုလမ်းမှာ တခြားလူတစ်ယောက်ကနိုးယူပြီး Data တွေပြင်ပေးလိုက်သလား၊ ဒါမှမဟုတ် Connection တစ်နေရာမှာချို့ယွင်းချက်တွေရှိလို့ Data တွေတစ်စိတ်တစ်ပိုင်းများ Lost ဖြစ်နေသလားဆိုတာသိနိုင်တဲ့ သင်္ချာဆိုင်ရာတွက်ချက်မှု ပုံစံမျိုးလေးတွေပါ။ တွက်ချက်ပုံနည်းလမ်းကို လိုက်ပြီးအမျိုးအမည်အများကြီးရှိပါတယ်။ ဒီတော့ကား ဒီ Error Control ကိုကြည့်ပြီးတော့ Receiver ဘက်မှာ Data ဟာ Original Data ဟုတ်မဟုတ်ဆိုတာကို ချက်ချင်းသိနိုင်တာပေါ့။ တကယ်လို့လက်ခံရရှိထားတဲ့ Packet ဟာ Original Packet မဟုတ်ဘူးဆိုရင်လဲ အဲဒီ Packet တစ်ခုတည်းကိုချက်ချင်း Sender ဘက်က Transport Layer ကထပ်ပြီးပို့ပေးပါတယ်။ ဒါကြောင့် Transport Layer ဟာ Data Transfer အစမှအဆုံးတိုင် Data Packet တစ်ခုချင်းစီအတိုင်းအတာအထိစိတ်ချရတဲ့ Data Transferring ဖြစ်စေဖို့အတွက် တာဝန်ယူလုပ်ဆောင်ပေးပါတယ်။ ဒါ့အပြင် Transport Layer နောက်ထပ်လုပ်ဆောင်ပေးတာကတော့ Data Transfer Speed ကို Control လုပ်ပေးတာပါပဲ။ Data Packets တွေကို Network Connection ပေါ်ကိုတင်ပို့ပေးတဲ့အခါမှာ ဟိုဘက်က လက်ခံမယ့် Receiver ဘက်က Switch တွေ၊ Network Card တွေကလက်ခံနိုင်မယ့် Speed အတိုင်းလိုက်လျောညီထွေဖြစ်အောင် ထိန်းချုပ်ပေး ပါတယ်။ ဒါပေမယ့် အဲဒီ Connection က Direct Connection (Sender နဲ့ Receiver တို့ဟာ Network တစ်ခုတည်းအတွင်းမှာပဲ ရှိနေရင်လို့ဆိုလိုတာပါ) ဆိုရင်တော့ ဒီကိစ္စဟာ မသိသာပါဘူး။ အဲဒါပေမယ့် Indirect Connection (Sender နဲ့ Receiver ကြားမှာ Network တွေအများကြီး ခြားရင်ခြားနေမယ်၊ ဒါမှမဟုတ် Internet ပေါ်ကဖြတ်သွားရတာမျိုးလည်းဖြစ်ရင်ဖြစ်နေမယ်၊ ဆိုလိုတာက တော့ Sender ကနေ Receiver ဆီကိုသွားမယ့် Data Packets တွေဟာကြားမှာ Router တို့ Gateway တို့လို High Performance Networking Device တွေအဆင့်ဆင့်ကို လက်ဆင့်ကမ်းဖြတ်ကျော်သွားရတာမျိုးကိုပြောတာပါ။) ဆိုရင်တော့ Transport Layer ရဲ့ လုပ်ဆောင်ချက်ကအရေးပါပါတယ်။ သူဟာလမ်းတစ်လျှောက်လုံးမှာ Data Packets တွေကိုဘယ်နေရာကနေဘယ်နေရာ

အတိုင်းအတာအထိကိုတော့ဘယ်လောက် Speed နဲ့သွားရမယ်၊ဘယ်နေရာကျရင်တော့ဘယ်လောက် Speed နဲ့သွားရမယ်ဆိုတာကို မှတ်သားလေ့လာဆုံဖြတ်ပြီးတော့ ထိန်းချုပ်ပေးပါတယ်။ ဒါကို Flow Control လုပ်ဆောင်တယ်လို့ခေါ်ပါတယ်။ဒါမှသာ Data တွေဟာ ကြားထဲမှာ Bottle Neck ပြဿနာလို တစ်ခုနဲ့တစ်ခု အထပ်ထပ် ထပ်မံပြီး ပျောက်ဆုံးပျက်ဆီးသွားတာတို့၊Packetsတစ်ခုနဲ့တစ်ခုကြား Time Limits တန်ဖိုးတစ်ခုလောက်အထိကွာပေးအဆက်ပြတ်သွားတာတို့ မဖြစ်နိုင်တော့ဘူးပေါ့နော်။ ဒါကြောင့် Transport Layer ရဲ့ လုပ်ဆောင်မှုတွေကိုအနှစ်ချုပ်ပြီး Reliable End-to-End Error Recovery and Flow Control လုပ်ပေးနိုင်တယ်လို့ပြောနိုင်ပါတယ်။ ဒီလောက်ဆိုရင်တော့ Transport Layer အကြောင်းကို တော်တော်လေး သဘောပေါက်လောက်ပြီလို့ထင်ပါတယ်။

ကဲ ကျွန်တော်တို့ နောက်ထပ် Layer တစ်ခုကို ထပ်ပြီး လေ့လာကြည့်ရအောင်။ ကျွန်တော်တို့အခုလေ့လာမယ့် Layer ကတော့ Network Layer (Layer 3) ပဲဖြစ်ပါတယ်။ အရင်ဆုံးကျွန်တော်တို့တစ်တွေ Network Layer ကဘာလဲဆိုတာကစပြီးစဉ်းစား ရအောင်။ မြန်မာလိုပဲတွေးကြည့်ဗျာ။ Network ဆိုတာကွန်ယက်တွေ၊ ကွန်ပျူတာတွေ တစ်လုံးနဲ့တစ်လုံးချိတ်ဆက်ထားတယ်ဆိုတဲ့ အဓိပ္ပာယ်ဟုတ်လား။ အဲဒီလိုပဲ Network Layer ဟာလည်း Sender နဲ့ Receiver ကြားမှာ Connection အဓိကချိတ်ဆက်ပေးတဲ့အလုပ် ကိုလုပ်ဆောင်ပါတယ်။ ဘယ်လို လုပ်ဆောင်ပေးတာလဲ။ ဟုတ်ကဲ့။ မတူညီတဲ့ Network တွေကြားမှာ ချိတ်ဆက်ပေးတာပါ။ (ဥပမာ Routers တို့ Gateway တို့လိုပေါ့။) Network Layer ရဲ့လုပ်ဆောင်မှုဟာလည်းအပေါ်က Transport Layer ရဲ့ Flow Control လုပ်ဆောင်မှုလိုပဲ Direct Connection တွေမှာတော့ သိပ်မသိသာ သိပ်အရေးမပါလှပါဘူး။ But ဒါပေမယ့်ပေါ့နော် Indirect Connection တွေမှာတော့ သိပ်ပြီးအရေးပါလှပါတယ်။ မတူညီတဲ့ Network ဆိုတာမျိုးကဘာကိုပြောတာလဲဆိုတော့နှစ်မျိုးရှိပါတယ်။ ပထမ တစ်မျိုးက Network Architecture တူပြီး Network Address တွေ Class မတူတာ Subnet မတူတာမျိုးပါ။ နောက်တစ်မျိုးက Network Architecture ကိုမတူတာပါ။ ပထမတစ်မျိုးမှာ အရင်ကြည့်ရအောင်။ ကျွန်တော်တို့တွေအနေနဲ့ IP Address တွေ အကြောင်းကို အနည်းငယ် သိထားဖို့တော့လိုပါမယ်။ IP Address ဆိုတာက TCP/IP Protocol ကိုသုံးတဲ့ Network တွေမှာ ကွန်ပျူတာတွေကို သတ်မှတ်ထားတဲ့ Logical Address တွေပါ။ တစ်လုံးနဲ့တစ်လုံး လုံးဝတူလို့မရပါဘူး။ အဲဒီ IP Address တွေမှာ အဓိက အသုံးပြုတာ Class (၃) မျိုး ရှိပါတယ်။ Class (a)ခုအတွင်းမှာ အကျုံးဝင်တဲ့ Address တွေနဲ့တခြား Class တွေမှာ အကျုံးဝင်တဲ့ Address တွေဟာ Network မတူပါဘူး။ Class တစ်ခုအတွင်းမှာ အကျုံးဝင်နေတဲ့ Address တွေကိုတောင် သီးခြားအစု လိုက်အစုလိုက် Network လေးတွေပြန်ခွဲနိုင်ပါသေးတယ်။ Subnet ကွဲတယ်လို့လဲပြောနိုင်ပါတယ်။(အသေးစိတ်ကိုတော့ IP Subnetting မှာလေ့လာလိုက်ပါနော်။ကျွန်တော်ကတော့ လိုသလောက်ပဲပြောသွားပါမယ်။) အဲဒီတော့ အဲဒီလိုမတူညီကြတဲ့ Network တွေဟာ Cable တွေနဲ့ဆက်သွယ်ထားပေမယ့် တစ်ခုနဲ့တစ်ခု Connection မရနိုင်ပါဘူး။ Data Transfer မလုပ်နိုင်ဘူးပေါ့ဗျာ။ အဲဒီလို အခါမျိုးမှာ Network Layer ကဝင်ပြီးတော့ သူတို့တွေကြားမှာ ချိတ်ဆက်ပေးပါတယ်။ ဒါမှလဲ သူတို့အချင်းချင်း Data Transfer လုပ်လို့ရသွားပါတယ်။ အဲဒါကိုကျွန်တော်တို့အနေနဲ့ Routing လုပ်တယ်လို့ခေါ်ပါတယ်။ နောက်တစ်မျိုးဖြစ်တဲ့ Network Architecture ကိုမတူတဲ့ Network တွေကြားမှာလည်း Network Layer ဟာသူတို့အချင်းချင်းကြားမှာဝင်ရောက်ချိတ်ဆက်ပေးပါတယ်။ Data Transfer လုပ်လို့ရနိုင်အောင်လို့ပါ။ အဲဒီမှာ တစ်ခုတော့ ပြောချင်ပါတယ်။ သူက Connection ရအောင်ပဲလုပ်ပေးတာပါ။ အဓိက ကတော့ Data Packet တွေကို ပို့နိုင်အောင် လက်ခံနိုင်အောင် သူကကြားဝင်တံတားထိုးပေးတဲ့သဘောမျိုးပါပဲ။ ပြောရရင်တော့ Only for Routing and Data Transmission ပါပဲ။ တခြားလိုအပ်တဲ့ Data ကို Processing လုပ်နိုင်အောင် နားလည်အောင် လုပ်ဆောင်ပေး ရတာတွေကတော့ အပေါ်ပိုင်း Layer တွေရဲ့ တာဝန်ပါ။သူနဲ့မဆိုင်ပါဘူး။

Network Layer ရဲ့အရေးပါတဲ့နောက်ထပ်လုပ်ဆောင်မှုတစ်ခုရှိပါသေးတယ်။ ခုနက ကျွန်တော်ပြောခဲ့သလိုပဲမတူညီတဲ့ Network တွေကြားမှာ ဝင်ရောက်တံတားထိုးပေးတဲ့သဘော Network Layer ကလုပ်ဆောင်ပါတယ်။ အဲဒီလို လုပ်ဆောင်တဲ့နေရာမှာ တံတားတစ်စင်းပဲထိုးလို့ရတယ်။ တခြားနေရာကသွားစရာလမ်းမရှိတော့ဘူး။ဒီတစ်လမ်းပဲရှိတော့တယ်ဆိုရင်တော့သိပ်မသိသာပါဘူး။ ဒါပေမယ့် တခြားသွားလို့ရတဲ့လမ်းတွေလည်းရှိတယ်။ အဲဒီလမ်းတွေဟာလည်း Data Packets တွေကို Receiver (Destination )ဆီကို

ရောက်စေနိုင်တယ်ဆိုရင် အဲဒီလမ်းတွေမှာလည်းတံတားတွေထိုး လိုက်ပြီး Data Packets တွေကို တစ်လမ်းတည်းက တစ်စုတစ်ဝေးတည်းမသွားစေတော့ဘဲ အဖွဲ့တွေခွဲပြီး တခြားလမ်းတွေကပါဖြန့်ပြီးသွားခိုင်းပါတယ်။ ဒါဟာ Data Transfer Speed ကိုအများကြီး တိုးတက်စေပါတယ်။ အဲဒါမျိုးဟာ Internet လိုအများသုံး Network ကြီးတွေ (လမ်းကြောင်းတွေအများကြီးအပြန်အလှန်ချိတ်ဆက်ထားကြတဲ့ Network အစုအဝေးတွေမှာပေါ့။) ပေါ်ကနေ Data Transfer လုပ်တဲ့အခါမှာ သိပ်အရေးပါပါတယ်။ ဒါပေမယ့်အဲဒီလို လမ်းကြောင်းတွေခွဲတဲ့အခါမှာ Network Layer ဟာရောက်ရင်ပြီးရောဆိုပြီးတော့ ပြီးပြီးရောပို့တာမျိုးတော့မဟုတ်ပါဘူး။ အရင်ဆုံး Sender ဘက် Network Layer ကသူနဲ့ချိတ်ဆက်ထားသမျှ Network လမ်းကြောင်းတွေတစ်လျှောက် Message တစ်ခုစီကို Time Limit တစ်ခုနဲ့ ချိန်ပြီးပို့လိုက်ပါတယ်။ အဲဒီ Message တွေထဲကမှ Time Limit မကုန်မီရောက်လာတဲ့ Message တွေကိုတော့ လာရာလမ်းအတိုင်း Receiver ဘက် Network Layer ကပြန်ပြီး အကြောင်းကြားစေပါတယ်။ Acknowledgement ပြန်တယ်ပေါ့ဗျာ။ Time Limit ကုန်တဲ့ အထိ Receiver ဘက်ကိုမရောက်သေးတဲ့ Message တွေကိုတော့ Time Limit ကုန်ပြီဆိုတာနဲ့ သက်ဆိုင်ရာ ရောက်ရှိနေတဲ့ Network ကသူဘာသာ Auto ဖျက်ချလိုက်ပါတယ်။ Sender ဘက်ကိုလည်း ပြန်အကြောင်းကြားမှု ရောက်မလာတော့ဘူးပေါ့။ အဲဒီလို ဖြစ်ပြီဆိုတော့မှ ခုနက အကြောင်းကြားမှုပြန်ရတဲ့လမ်းကြောင်းတွေအတိုင်း Network Layer ကလိုက်ပြီး တံတားထိုး (Route လုပ်တယ်ပေါ့ဗျာ) ပါတယ်။ ပြီးတာနဲ့ Data Transfer လုပ်နိုင်ဖို့ အောက်ပိုင်း Layer တွေဆီ လက်ဆင့်ကမ်းပို့ဆောင်ပါတယ်။ ကဲဒီလောက်ဆိုရင် Network Layer ရဲ့လုပ်ဆောင်ပုံတွေကို နားလည် သဘောပေါက်လောက်ပြီ လို့ထင်ပါတယ်။

ကဲနောက်တစ်ဆင့်ကတော့ Data Link Layer (Layer 2) ပါ။ Data Link Layer ရဲ့အဓိကလုပ်ဆောင်ချက်ကတော့ အပေါ် Layer တွေဆီကလက်ခံရရှိလာတဲ့ Data Packet လေးတွေကို Network Connection တစ်လျှောက်မှာ သွားလာနိုင်မယ့် Data Frame အဖြစ်ပြောင်းလဲပေးပါတယ်။ အဲဒီ Data Frame ပေါ်မှာလိုအပ်တဲ့ Information တွေလည်းထပ်ထည့်ပါတယ်။ ပြီးတော့ Data တွေ ကိုလည်း အောက်က Physical Layer ကလက်ခံနားလည်ပြီး Network Connection ပေါ်တင်ပို့ပေးနိုင်အောင်လို့ Raw Bits တွေအဖြစ်ပြောင်းလဲပေးပါတယ်။ Data Link Layer ရဲ့အဓိကနောက်ထပ်လုပ် ဆောင်ချက် တစ်ခုကတော့ Sender နဲ့အနီးစပ်ဆုံး လက်ခံမယ့် Receiver (Destination Receiver အစစ်မဟုတ်သေးတဲ့ကြားကလက်ဆင့်ကမ်းပေးမယ့်သူကိုပြောတာပါ) ကြားထဲက Data သွားမယ့် Network Connection ဟာ Clear ဖြစ်နေလား၊ Data သွားမယ်ဆိုရင် Collision ဖြစ်နိုင်လားဆိုတာတွေကို လေ့လာစစ်ဆေးပြီး စိတ်ချရပြီဆိုမှ Data တွေကိုပို့လွှတ်စေပါတယ်။ နောက်တစ်ပိုင်းဖြစ်တဲ့ ခုနကလက်ခံခဲ့တဲ့ Receiver နဲ့ နောက်တစ်ဆင့် Receiver ကြားမှာလည်းအဲဒီလိုစစ်ဆေးပြီး စောင့်သင့်ရင်စောင့်ပြီး စိတ်ချရမှ ပို့စေပါတယ်။အဲဒီလိုလုပ်တာဟာ တကယ့် Destination Receiver အစစ်ဆီကို ရောက်တဲ့အထိပါပဲ။ အဲဒီလိုစစ်ဆေးတဲ့နေရာမှာ ကြားခံ Network ရဲ့သုံးထားတဲ့နည်းပညာပေါ်ကို မူတည်ပြီး စစ်ဆေးပုံစစ်ဆေးနည်းတွေကတော့အများကြီးရှိပါတယ်။ Data Link Layer ကိုထပ်ပြီး (၂)ပိုင်း ပိုင်းနိုင်ပါသေးတယ်။ Logical Link Control (LLC) ဆိုတဲ့ Sublayer နဲ့ Media Access Control (MAC) ဆိုတဲ့ Sublayer တို့ပဲဖြစ်ပါတယ်။ Logical Link Control (LLC) ကတော့ ခုနကပြောခဲ့တဲ့ Collision မဖြစ်အောင်စောင့်ထိန်းပေးတဲ့ကိစ္စမှာ အရင်ဆုံး Data တွေကို မပို့ခင် Nodes နှစ်ခုကြားမှာ (Sender နဲ့အနီးဆုံး Receiver ပဲထားပါတော့ဗျာ) Cable တွေ Card တွေကို Connection ရအောင် လုပ်ပါတယ်။ Logical Link လေးပေါ့ဗျာ။အဲဒီ Link ပေါ်မှာ သွားနိုင်အောင် Data တွေကို အဆင်သင့်ဖြစ်အောင်ပြင်ဆင်ပေးပါတယ်။ Raw Bits တွေဖြစ်လာအောင်ပေါ့။ ဘယ်အပိုင်းက အရင်သွားရမယ်၊ ဘယ်အပိုင်းက နောက်ကလိုက်မယ်ဆိုတာကိုလဲ ဆုံးဖြတ်ပေးပါတယ်။ လက်ခံမယ့်ဘက်မှာလည်း Data တွေကို အပေါ် Layer တွေဆီ တင်ပေးနိုင်ဖို့အဆင်သင့်အနေအထားဖြစ်အောင် ပြန်ပြုပြင်ပါတယ်။ Data Frame တွေကိုလည်း အစီအစဉ်တကျပြန်ပြီးစီရတာပေါ့ဗျာ။ အဲနောက်တစ်လွှာဖြစ်တဲ့ Media Access Control ကတော့ LLC Sublayer နဲ့ Physical Layer ကြားမှာ Interface ပုံစံမျိုးလုပ်ပေးပါတယ်။ LLC Sublayer ဆီက Data Frame တွေကို တစ်ခုချင်းစီရယူပြီး Data Collision ဖြစ်နိုင်မဖြစ်နိုင်စောင့်ကြည့်ကာ မဖြစ်နိုင်ဘူးဆိုရင် Transmit လုပ်ပါတယ်။ Collision ဖြစ်နိုင်တယ်ဆိုရင်ယာယီစောင့်နေ ပါသေးတယ်။ ပြီးတော့လည်း Data Frame တစ်ခုပြီးမှတစ်ခုယူပြီး Transmit လုပ်တာပါ။ လက်ခံမယ့်ဘက်ကတော့ တစ်ခုချင်းစီလက်ခံစစ်ဆေးပြီးပါလာတဲ့ Information တွေကိုကိုညီတယ်ဆိုမှ LLC Sublayer



ကိုလက်ဆင့်ကမ်းပေးပါတယ်။ အကယ်၍ ပါလာတဲ့ Information တွေနဲ့ မကိုက်ညီဘူးဆိုရင် အဲဒီ Frame ကိုထပ်ပို့ခိုင်းပါတယ်။ Data Link Layer နဲ့သူ့ရဲ့ Sublayer နှစ်ခုအကြောင်းကို ကောင်းစွာသဘောပေါက် လောက်ပြီလို့ထင်ပါတယ်။

ကဲလာပါပြီ။ နောက်ဆုံး Layer 1 ဖြစ်တဲ့ Physical Layer ပါ။ Physical Layer ကဘာတွေလုပ်ပါသလဲ။ Physical Layer ကတော့ Network Cable ပေါ်မှာ Data တွေကို အမှန်တကယ် Transmission လုပ်နိုင်အောင်လုပ်ဆောင်ပေးပါတယ်။ အပေါ် Data Link Layer ကတဆင့်ရရှိလာတဲ့ Raw Bits Data Frame တွေကို Cable တွေပေါ်မှာ အမှန်တကယ်သွားနိုင်မယ့် Data Signals များအဖြစ်သို့ပြောင်းလဲပေးပါတယ်။ ပြောင်းလဲတဲ့နေရာမှာ Cable အမျိုးအစားတွေ၊ Card အမျိုးအစားတွေ၊ Connector အမျိုးအစားတွေကိုလိုက်ပြီး တစ်ဘက်နဲ့တစ်ဘက် Transfer လုပ်လို့အဆင်ပြေနိုင်မယ့် Signal မျိုးကိုပြောင်းလဲပေးပါတယ်။ Cable တွေ၊ Card တွေ၊ Connector တွေအမျိုးအစား အများကြီးရှိသလို Physical Layer ကပြောင်းလဲပေးရတဲ့ Signal အမျိုးအစားတွေလည်းအများကြီးတော့ ရှိတာပေါ့နော်။ ဥပမာ Digital Signal ၊ Analog Signal တို့ပေါ့။ ပြီးတော့ Signal အမျိုးအစားတစ်ခုချင်းစီအတွက်တောင် ကြားထဲက ချိတ်ဆက်ထားတဲ့ Network Connection ရဲ့နည်းပညာပေါ်မူတည်ပြီး ထုတ်လွှတ်ပုံထုတ်လွှတ်နည်းတွေကွဲပြားပါသေး တယ်။ (ဥပမာ Star လား Ring လား Bus လား) ဒါကြောင့် Physical Layer ဟာတကယ့် Really Physical Media ပေါ်မှာ Data တွေမှန်မှန်ကန်ကန် အဆင်ပြေပြေသွားနိုင်ဖို့အတွက်ကို လိုအပ်တဲ့ လုပ်ဆောင်မှုတွေဖြစ်တဲ့ Raw Bits Data Frame တွေကို Electric Signal အသွင်ပြောင်းပေးတာတွေ၊ အဲဒီလိုပြောင်းတဲ့အခါမှာ ဘယ်လို Signal ပုံစံကို ပြောင်းမှာလဲဆိုတာတွေ၊ ပြောင်းပြီးသား Signal တွေကို ဘယ်လိုဘယ်ပုံ Cable ကြိုးပေါ်ကို တင်ပေးမှာလဲ ဆိုတာတွေအပြင် Cable ပေါ်ကို Data တွေတင်ပေးနိုင်ဖို့အတွက် လိုအပ်တဲ့ Voltage တွေကအစဆုံးဖြတ်သတ်မှတ်ထုတ်လုပ်ပေးတာကိုလုပ်ဆောင်ပါတယ်။ ကဲဒီလောက်ဆိုရင်တော့ Physical Layer ရဲ့အကြောင်းကိုလည်းတော်တော်သဘော ပေါက်လောက်ပြီလို့ထင်ပါတယ်။

ကဲဒါဆိုရင် Layer (၇)ခု လုံးအကြောင်းကို အားရပါးရ စိတ်ဝင်တစားဖတ်ပြီးသွားတဲ့အခါ ခေါင်းထဲမှာ အရေးကြီးတာလေးတွေကို စွဲကျန်မှတ်မိနေအောင်လို့ Layer (၇) ခုလုံးကို အနှစ်ခြုပ်ပြီးပြန်လည်ဖော်ပြလိုက်ပါတယ်။ ဒီတစ်ခါတော့ Network အခေါ်အဝေါ် Layer နံပါတ်စဉ်အတိုင်းဖော်ပြလိုက်ပါတယ်နော်။

### Physical Layer

Transfer Medium ပေါ်မှာ Data သွားလာနိုင်ဖို့ ကူညီပေးပါတယ်။ Data တွေကို Transfer Medium နဲ့လိုက်လျောညီထွေရှိမယ့် Signal ပုံစံအဖြစ်ပြောင်းလဲပေးပါတယ်။ အဲဒီ Data Signal တွေကို Transmission လုပ်နိုင်ဖို့ရောပြန်ပြီး Synchronization လုပ်နိုင်ဖို့ရော လိုအပ်တဲ့ Voltage Levels ကိုဆုံးဖြတ်ပေးပါတယ်။

### Data Link Layer

Network လမ်းကြောင်းတစ်လျှောက်မှာအဆင်ပြေပြေသွားနိုင်မယ့် Frame လေးတွေဖြစ်လာအောင် Data ကိုတည်ဆောက်ပေးပါတယ်။ Network Connection တစ်လျှောက်မှာCollision မဖြစ်အောင် စောင့်ကြပ်ထိန်းသိမ်းပေးပါတယ်။ Collision ဖြစ်သွားရင် ဒါမှမဟုတ် Error Control Information နဲ့မတိုက်ဆိုင်ပဲ Error တွေနေရတယ်ဆိုရင် အဲဒီFrame ကိုချက်ချင်းထပ်ပို့ပေးပါတယ်။

### Network Layer

မတူညီတဲ့ Network တွေကြားမှာ Connection ရအောင် ၊ Data Transfer လုပ်နိုင်အောင်ကြားခံဆက် သွယ်ပေးပါတယ်။ Network လမ်းကြောင်းတွေကိုလည်းရှာဖွေပေးပြီးData Transferလုပ်ရာမှာပိုမိုမြန် ခဆန်အောင် ကူညီပေးပါတယ်။

#### Transport Layer

Data Transfer လုပ်ရာမှာအစမှအဆုံးတိုင် Data Packet တစ်ခုချင်းစီအတိုင်းအတာအထိ စိတ်ချရတဲ့ Data Transmission တစ်ခု ဖြစ်အောင်လုပ်ဆောင်ပေးပါတယ်။ Error Control နဲ့ Flow Control ကို ဆောင်ရွက်ပါတယ်။ အကယ်၍ Data Packet တစ်ခုမှာများနဂို Packetအတိုင်းမဟုတ်ကြောင်း တွေ့ရှိပါကချက်ချင်းထပ်ပို့ပေးပါတယ်။

#### Session Layer

Data Transfer လုပ်ဖို့အတွက် Sender နဲ့ Receiver ကြားမှာ Logical Connection တစ်ခုကို အစပြုတည်ဆောက်ပါတယ်။ အဲဒီ Connection ကို Data Transmission မပြီးမချင်းထိန်းသိမ်းထားပါတယ်။ Data တွေကိုလည်းအဲဒီ Connection ပေါ်မှာ အဆင်ပြေ ပြေ သွားနိုင်မယ့်အပိုင်းလေးတွေအဖြစ်ပိုင်းပါတယ်။ လက်ခံမယ့်ဘက်ရောက်တဲ့အခါမှာနဂိုမူရင်း Original Data ပြန်ရအောင် ပြန် လည်စုစည်းပါတယ်။

#### Presentation Layer

Data တွေကိုအောက်ပိုင်း Layer တွေကနားလည်အဆင်ပြေမယ့် Format ပုံစံမျိုးပြောင်းလဲပေးပါတယ်။ လိုအပ်တဲ့ Encryption တွေနဲ့ Compression တွေကိုလည်းလုပ်ဆောင်ပေးပါတယ်။ လက်ခံမယ့်ဘက်မှာကျတော့ Decompression တွေနဲ့ Decryption တွေကိုလိုအပ်သလို လုပ်ဆောင်ပါတယ်။ User နားလည်လက်ခံနိုင်မယ့် Format မျိုးပြန်ရအောင် Data ကိုပြန်ပြီး Format ပြောင်းပေးပါတယ်။

#### Application Layer

User တွေပို့ချင်တဲ့ သတင်းအချက်အလက်တွေ၊ Data တွေကို အောက်ပိုင်း Layer တွေက နားလည်လက်ခံပြီးဆက်လက် Process လုပ်နိုင်ဖို့အတွက် Interface တစ်ခုအဖြစ်ကြားခံပြီး User တွေကို ပိုမို လွယ်ကူအဆင်ပြေစေပါတယ်။ လက်ခံမယ့်ဘက်မှာလည်း User တွေအနေနဲ့လက်ခံရရှိလာတဲ့ Data တွေကို အသုံးပြုချင်သလိုအသုံးပြုနိုင်ခွင့် ရရှိအောင် Interface တစ်ခုအနေနဲ့ Userတွေကို ပိုမိုအဆင်ပြေစေရန် အသုံးတော်ခံပြန်ပါတယ်။