**Message Format:** "Message Format" veya "Mesaj Formatı", iletişim protokolleri veya veri alışverişi formatları gibi sistemlerde kullanılan standart bir mesaj yapısıdır. Bu yapının amacı, farklı sistemler arasında veri alışverişi yaparken, verilerin birbirlerini anlamasını sağlamaktır.

Genellikle, bir mesaj formatı, başlık, veri alanı ve sonlandırma karakterleri gibi belirli öğelerden oluşur. Başlık bölümü, mesajın türü, mesajın boyutu, mesajın kaynağı ve hedefi gibi bilgileri içerebilir. Veri alanı bölümü, mesajın kendisini içerir ve aktarılması gereken bilgileri taşır. Sonlandırma karakterleri, mesajın sonlandırıldığını belirtir.

Mesaj formatları, farklı uygulama alanlarına veya iletişim protokollerine göre değişebilir. Örneğin, bir e-posta mesajı için mesaj formatı, başlık, gövde ve ekler gibi farklı bölümler içerebilirken, bir HTTP isteği için mesaj formatı, isteğin yöntemi, URI, sürümü ve başlık alanları gibi farklı bileşenlerden oluşabilir.

Mesaj formatları, veri alışverişi yapan farklı sistemler arasında uyumlu ve güvenilir bir şekilde veri alışverişi yapmak için kullanılır.

**Message Size :** "Message Size", bir mesajın boyutunu ifade eder. Bu, bir mesajda iletilen toplam veri miktarıdır.

Mesaj boyutu, mesajı taşıyan iletişim protokolüne, veri alışverişi formatına ve uygulama senaryosuna bağlı olarak değişebilir. Örneğin, e-posta mesajları genellikle büyük boyutlarda olabilir, çünkü birçok farklı bileşen içerebilirler, örneğin gövde, başlık, ekler vb. HTTP istek ve yanıt mesajları, genellikle daha küçük boyutlarda olabilir, çünkü yalnızca belirli bilgileri taşımaları gerekir.

Mesaj boyutu, veri aktarımında önemli bir faktördür. Özellikle, düşük bant genişlikli veya yavaş bağlantılar gibi sınırlı kaynaklara sahip sistemlerde, büyük mesajlar daha uzun sürelerde aktarılır ve performansı düşürebilir. Bu nedenle, mesaj boyutunun mümkün olduğunca küçük tutulması, daha hızlı ve verimli bir veri alışverişi için önemlidir.

**Encoding** "Encoding", bilgi veya verilerin belirli bir format veya yöntem kullanarak kodlanması işlemidir. Encoding, verilerin taşınabilir hale getirilmesi veya saklanması için kullanılır ve verilerin iletimi, depolanması veya işlenmesi sırasında hata olasılığını azaltır.  
**Encapsulation** Encapsulation, bir nesnenin veya veri yapısının işlevselliğinin ve güvenliğinin korunmasına yardımcı olan bir nesne yönelimli programlama (OOP) prensibidir. Encapsulation, nesnenin iç durumunu ve davranışlarını gizlemek için kullanılır ve bu sayede nesnenin kullanıcısı sadece gerekli olan bilgileri görebilir ve manipüle edebilir. **Message pattern** "Message pattern" (mesaj şablonu), yazılım mimarisi ve tasarımında, farklı bileşenler arasında mesajların nasıl gönderildiği ve nasıl işlendiği konusunda bir planlama ve tanımlama yöntemidir.

Bir mesaj şablonu, bir sistemdeki bileşenler arasındaki iletişimde kullanılan mesajların türünü, formatını, içeriğini ve mesajların hangi durumlarda gönderileceğini tanımlar. Bu, bileşenlerin nasıl birbirleriyle iletişim kuracağını ve veri paylaşacaklarını belirleyen standart bir iletişim protokolü sağlar. **Standards Organizations** Standards Organizations" (Standartlar Örgütleri), endüstride veya belli bir sektörde ortak standartlar belirlemek ve yönetmek amacıyla kurulmuş kuruluşlardır. Bu kuruluşlar, belirli bir sektördeki tüm paydaşların (firmalar, hükümetler, akademik kuruluşlar, tüketici grupları vb.) katılımıyla standartlar oluştururlar ve bunların uygulanması ve takibi konusunda denetim yaparlar. **TCP/IP Model** TCP/IP modeli, internetin temel iletişim protokollerini belirleyen bir ağ protokolüdür. Bu model, İnternet Protokolü (IP) ve Transmission Control Protocol (TCP) gibi protokollerin yanı sıra, diğer çeşitli protokollerin de kullanımını destekler.

TCP/IP modeli, dört katmandan oluşur:

1. Ağ Erişim Katmanı (Network Access Layer): Bu katman, ağa fiziksel olarak bağlı cihazların ağa erişimini sağlar ve veri paketlerinin kablosal veya kablolusistemler üzerinden iletilmesini yönetir. Ethernet, Wi-Fi veya DSL gibi protokoller bu katmanda kullanılır.
2. İnternet Katmanı (Internet Layer): Bu katman, IP protokolünü kullanarak veri paketlerinin gönderilmesini yönetir. Bu katmanda, veri paketleri adreslenir ve yönlendirilir. IP protokolü, tüm internet trafiği için temel protokol olarak kabul edilir.
3. Taşıma Katmanı (Transport Layer): Bu katman, veri iletimi için güvenilir bir bağlantı sağlar ve veri bütünlüğünü ve güvenliğini sağlamak için TCP protokolünü kullanır. TCP, veri akışını düzenler, hata düzeltmeleri yapar ve paket kayıplarını tespit eder.
4. Uygulama Katmanı (Application Layer): Bu katman, ağ uygulamaları ve servisleri için iletişim kurallarını belirler. HTTP, FTP, SMTP ve DNS gibi birçok protokol bu katmanda kullanılır.

TCP/IP modeli, OSI (Open Systems Interconnection) modeline benzer bir yapıya sahip olmasına rağmen, daha basit ve esnek bir yapı sunar. TCP/IP, internetin en temel protokolüdür ve bugün hala kullanılan en yaygın ağ protokolüdür.

**http** HTTP (Hyper Text Transfer Protocol) bir ağ protokolüdür ve internet üzerinden iletişim kurmak için kullanılır. HTTP, web sayfaları, resimler, videolar ve diğer web içeriği gibi birçok farklı türdeki veriyi aktarmak için kullanılır.

HTTP, istemci-sunucu modeline dayalıdır. Web tarayıcılar, web sayfasını isteyen istemci olarak hareket ederken, web sunucuları, istekleri karşılayan sunucu olarak hareket eder. İstemci, sunucuya bir istek gönderir ve sunucu, isteği karşılayarak yanıt verir.

**TCP** TCP (Transmission Control Protocol), internet üzerinden veri transferi yapmak için kullanılan bir ağ protokolüdür. TCP, ağda doğru ve güvenilir bir şekilde veri transferi yapmak için paket yönlendirmesi, paket yeniden gönderimi, paket sıralaması ve hata kontrolü gibi özellikleri sağlar.

TCP, istemci-sunucu modeliyle çalışır. İstemci, bir sunucu ile bağlantı kurmak için bir istek gönderir. Sunucu, isteği aldıktan sonra bir yanıt gönderir ve böylece bir bağlantı kurulur. TCP bağlantıları, ağda birçok farklı aygıt ve ağ protokolü arasında sağlam ve güvenilir bir bağlantı oluşturmak için kullanılır.