

Şadi Evren Şeker & Rabia Yörük

İNTERNET NASIL ÇALIŞIR?

*Bu kitabın bütün hakları saklıdır ve
hakları yazarlarına aittir.*

İÇİNDEKİLER

İÇİNDEKİLER.....	1
ÖNSÖZ.....	5
1.BÖLÜM: AĞ TEORİSİNE GİRİŞ.....	6
1.1 AĞ TEORİSİNE GİRİŞ	7
1.2 AĞ TEORİSİ NEDİR?	7
1.3 İLETİŞİMİN GEÇMİŞİ.....	8
1.4 İNTERNETİN TARİHİ.....	9
2.BÖLÜM: AĞ TEORİSİ.....	10
2.1 AĞ TEORİSİ – TEMEL KAVRAMLAR.....	11
2.1.1 DİJİTAL SİSTEMLER.....	11
2.2 AĞ TAŞINMASI.....	12
2.2.1 UTP	12
2.2.2 BNC.....	13
2.2.3 FİBER OPTİK.....	13
2.3 VERİLERİN DÖNÜŞÜMÜ	14
2.4 VERİNİN ULAŞTIRILMASI.....	16
2.5 AKTARIM HIZINI ETKİLEYEN FAKTÖRLER.....	16
.....Hata! Yer işareti tanımlanmamış.	
2.6 GENİŞ BANDI ETKİLEYEN FAKTÖRLER	18
2.7 HESAPLAMALAR.....	20
2.7.1 VERİNİN İLETİM SÜRESİNİN HESAPLANMASI.....	20
2.7.1 VERİNİN İLETİM HIZININ HESAPLANMASI.....	20
3.BÖLÜM: İNTERNET PROTOKOLÜ,.....	21
DNS VE ALAN İSİMLERİ	21
3.1 İNTERNET NEDİR?	22
3.2 İNTERNET DONANIMI	23
3.3 İNTERNET YAZILIMI	24
3.4 PAKET ANAHTARLAMA	25
3.4.1 PAKET ANAHTARLAMAMININ TARİHİ.....	25
3.4.2 PAKET ANAHTARLAMA NEDİR?	26
3.5 İNTERNET PROTOKOLÜ	27
3.5.1 IP ADRESİ NASIL ÖĞRENİLİR?	28
3.5.2 IP ADRESLERİNİN ULAŞILABİLİRLİĞİ.....	32
3.5.3 IP ADRESLERİNİN ÖZELLİKLERİ	34
3.6 DOMAIN İSİMLERİ.....	35
3.7 DNS (DIRECT NAME SERVER)	36
3.7.1 DNS NASIL ÇALIŞIR?	37
3.8 İNTERNET POLİTİKALARI.....	38
4.BÖLÜM: NAT	39
4.1 NAT HAKKINDA.....	40

4.2 NAT NASIL ÇALIŞIR?	41
5.BÖLÜM: AĞ PROTOKOLLERİ.....	43
5.1 AĞ STANDARTLARI.....	44
5.1.1 OSI (OPEN SYSTEM INTERCONNECT)	44
5.1.1.1 FİZİKSEL KATMAN	45
5.1.1.2 VERİ BAĞLANTI KATMANI.....	45
5.1.1.3 AĞ KATMANI.....	45
5.1.1.4 TAŞIMA KATMANI.....	45
5.1.1.5 OTURUM KATMANI.....	45
5.1.1.6 SUNUM KATMANI.....	46
5.1.1.7 UYGULAMA KATMANI	46
5.1.2 TCP/IP	47
5.1.2.1 LİNK KATMANI.....	48
5.1.2.2 AĞ KATMANI.....	48
5.1.2.3 UYGULAMA KATMANI	48
5.1.2.4 TAŞIMA KATMANI (TCP – TRANSPORT CONROL PROTOCOL).....	49
5.2 KATMANLAR VE PROTOKOLLER NASIL İŞLER?	49
..... Hata! Yer işareti tanımlanmamış.	
5.3 THREE WAY HANDSHAKE	50
5.4 DOS (DENIAL OF SERVICE ATTACKS) SALDIRILARI	51
5.5 DOSYA GÖNDERİMİ	52
6.BÖLÜM: WORLD WIDE WEB	54
6.1 WWW TARİHİ.....	55
6.2 BROWSER REKABETLERİ.....	56
6.3 WEB STANDARTLARI	58
6.3.1 HTML.....	58
6.3.2 CSS.....	58
6.4 URL (UNIFORM RESOURCE LOCATORS)	59
6.4.1 URL NASIL ÇALIŞIR?	59
6.5 İSTEMCİ – SUNUCU İLİŞKİLERİ	60
6.6 TELNET	61
6.6.1 TELNET KOMUTU NASIL ÇALIŞIR?.....	61
7.BÖLÜM: HTML	63
7.1 HTML NASIL ÇALIŞIR?.....	64
7.2 UYGULAMA KURULUMU (MAMP)	64
7.3 HTML TARİHİ.....	70
7.4 HTML NEDİR?.....	71
7.4.1 HTML KODLARI HAKKINDA ÖNEMLİ NOKTALAR	71
7.5 GENEL BAKIŞ	72
7.6 ETİKETLER (TAGS).....	76
7.7 HYPERLINK	77
7.8 KAÇIŞ KODLARI (ESCAPE CODES).....	79
7.9 HTML – LİSTELER	81

7.10 HTML – TABLOLAR.....	85
7. 11 RESİM FORMATLARI.....	99
8.BÖLÜM: CSS – STİLLER (STYLES).....	100
8.1 CSS – STİLLER (STYLES).....	101
9.BÖLÜM: HTML – FORMLAR.....	110
9.1 FORM NEDİR?.....	111
9.2 FORM NASIL OLUŞTURULUR?.....	112
10.BÖLÜM: E-POSTA (E-MAİL) NASIL ÇALIŞIR?.....	117
10.1 E-POSTA’NIN ÇALIŞMA DÖNGÜSÜ.....	118
10.2 SMTP (SIMPLE MAIL TRANSPORT PROTOCOL)	119
10.2.1 SMTP FORMATI.....	119
10.2.2 SAAT DİLİMLERİ (TIMESTAMP).....	120
10.2.3 E-POSTA EKLERİ (E-MAİL ATTACHMENTS)	120
10.3 POP (POST OFFICE PROTOCOL) /IMAP (INTERNET MAIL ACCESS PROTOCOL)	122
10.4 SPAM E-POSTALAR.....	123
11.BÖLÜM: İNTERNET GÜVENLİĞİ VE KİŞİSEL ÖNLEMLER	124
11.1 CHAINMAIL	125
11.2 URBAN LEGEND.....	125
11.3 PHISHING.....	126
11.4 PHISHING İÇİN ALINABİLECEK ÖNLEMLER	126
11.5 PHARMING	127
11.6 SERTİFİKALAR	127
11.7 KÖTÜ AMAÇLI YAZILIMLAR (MALWARE)	128
11.7.1 KÖTÜ AMAÇLI YAZILIM TÜRLERİ.....	128
11.7.2 KÖTÜ AMAÇLI YAZILIMLAR NASIL YAYILIR?.....	130
11.7.3 KÖTÜ AMAÇLI YAZILIMLARA KARŞI ALINABİLECEK ÖNLEMLER	130
12.BÖLÜM: İNTERNET ÜZERİNDE ARAMA NASIL YAPILIR?	131
12.1 VERİ KOLEKSİYONU (DATA COLLECTION)	132
12.2 ARAMA MODELİ.....	133
12.2.1 KELİME HAFIZASI.....	133
12.3 OTORİTE BAZLI ARAMA (AUTHORITY BASED SEARCH)	134
12.4 ETKİLİ ARAMA TEKNİKLERİ	137
13.BÖLÜM: ŞİFRELEME/KRİPTOLOJİ.....	140
13.1 KRİPTOLOJİ BİLİMİ (CRYPTOLOGY).....	141
13.2 SEZAR ŞİFRELEMESİ.....	141
13.3 ONE TIME PAD	142
13.4 ANAHTARLARIN DAĞITILMASI.....	143
13.4.1 Asimetrik Şifreleme İçin Kullanılan Anahtarlama	143
13.4.2 Simetrik Şifreleme İçin Kullanılan Anahtarlama	143
13.5 ŞİFRELERİN KIRILMASI.....	144

13.5.1 ENİGMA MAKİNESİ.....	144
13.6 ANAHTARLARIN DEĞİŞİMİ	146
13.7 KİŞİSEL ŞİFRELEME (PGP)	147
13.7.2 PGP'NİN ÇALIŞMA PRENSİBİ.....	147

ÖNSÖZ

Bu kitap, çağımızda büyük bir geçerliliğe sahip olan ve yaygın olarak kullanılan internetin mantığını kavrayabilmeniz amaçlanarak yazılmıştır.

İnternet nedir? Nasıl çalışır? Web sitesi nedir? Gibi soruların cevaplarını, mantığı ile birlikte sizlere aktarırken, aynı zamanda ağ teorisine de giriş yapacağız. Böylece vermiş olduğumuz her cevap, ağ teorisine uygun olarak cevaplanacaktır.

“İnternet Nasıl Çalışır”, ağ teorisini öğrenmek isteyen ve bu konuda istekli olan lise veya üniversite öğrencileri başta olmak üzere, herkese hitap edecek şekilde hazırlanmıştır. Dolayısı ile her seviyeye uygun olarak anlatım yapılmıştır.

Ayrıca kitabımız içerisinde kullanmış olduğumuz her kaynağa www.sadievrenseker.com adresi üzerinde, “kurslar (courses)” başlığı altından ulaşabilmeniz mümkündür.

1.BÖLÜM: AĞ TEORİSİNE GİRİŞ

1.1 AĞ TEORİSİNE GİRİŞ

1.2 AĞ TEORİSİ NEDİR?

1.3 İLETİŞİMİN GEÇMİŞİ

1.4 İNTERNETİN TARİHİ

1.1 AĞ TEORİSİNE GİRİŞ

Geçmişten günümüze doğru geldiğimiz zaman, insanlar birbirleri ile iletişim kurabilmek için birçok iletişim yöntemi kullanmışlardır. Duman, güvercin, mektup, telgraf, gazete bunlardan sadece birkaçıdır. Günümüze bakıldığında zaman ise birçok insanın artık iletişim ihtiyacını internet üzerinden sağladığı görülmektedir. Peki internet üzerinden sağladığımız iletişimde kullanılan yöntemler ile geçmişte kullanılan yöntemler arasında bir fark var mıdır?



Resim 1.1.1

1.2 AĞ TEORİSİ NEDİR?

Ağ teorisi, ayrı nesnelere arasındaki asimetrik ve simetrik ilişkiler üzerine kuruludur. Genel olarak grafiklerin incelenmesi üzerine kurulu bir sisteme sahiptir ve aynı zamanda bilgisayar ve ağ biliminde grafik teorisinin bir parçası olarak görülmektedir.

Ağ teorisi bilgisayar bilimi, parçacık fiziği, internet, dünya çapında ağ (www) vb. konular ile yakından ilişkilidir.

Karmaşık ağları karakterize etmek ve modelleme yöntemini incelemek için ağ teorisi sıklıkla kullanılır.

1.3 İLETİŞİMİN GEÇMİŞİ

İlk çağlardan itibaren insanlar iletişim kurmak için yeni formüller, icatlar ve keşifler yapmıştır. Önceleri insanlar birbirleri ile bağırarak haberleşirken, daha sonra daha uzak mesafelere seslerini duyurabilmek için davul kullanmışlar ve duman, güvercin, telgraf, matbaa, telefon ve son olarak internetin gelişmesi ile farklı iletişim yolları bulmuşlardır.

Bunlar arasından internet günümüzde en önemli iletişim ağlarından biri haline gelmiştir. Ve aslında bir iletinin bir yerden başka bir yere iletilmesi ile gerçekleşir. Bu kısımda interneti yalnızca iletişim aracı olarak düşünmemeliyiz. Örnek olarak eskiden yazılı olarak tutulan belgeler ve dolayısıyla bilgiler, şu an internette yer almaktadırlar. Eskiden toplantı, buluşma veya görüşme için toplanılan yerler, şu an internet üzerinden sağlanabilmektedir.

Günümüzde insanların işlerinin çoğunu internet üzerinden sağladıkları da göz önünde bulundurulursa, konunun sadece internet ile sınırlı kalmayıp aynı zamanda dijitalleşmenin de söz konusu olduğu ortaya çıkmaktadır. Örneğin çoğumuzun mobil uygulamalarını veya web sürümlerini kullandığı, e-ticaret siteleri veya hizmet sektöründe e-Devlet, e-SGK veya e-Demokrasi siteleri bunlara örnek olarak gösterebilir.

Özetlemek gerekirse geçmişte hizmet ve bilgi sağlamak amaçlı kullanılan sistemlerin çoğu, günümüzde internet üzerinden sağlanabilmektedir. Bunlara ek olarak bir iletişim aracı olarak da kullanılmaktadır ki, bizlere sistemlerin hepsini taşıyan ve bunlara ulaşabildiğimiz bir ortam sağlamaktadır.

1.4 İNTERNETİN TARİHİ

Her ne kadar günümüzde en çok bir iletişim aracı olarak kullanılsa da, internetin ilk icat edilme tarihi askeri bir projeye dayanmaktadır. Eskiden merkezi bir sisteme sahip olan Amerika'nın, Savunma bakanlığı tarafından başlatılan çalışmalar, nükleer bir savaş çıkma durumunda bile iletişimin devam edilebilmesi isteği üzerine geliştirilmiştir.

Gücü merkezde toplayan bir sisteme sahip olan ülkeler, ürettiği teknolojide de gücü toplayan bir sisteme sahip olurken; gücü dağıtan bir sisteme sahip olan ülkeler, ürettiği teknoloji itibari ile de gücü dağıtan bir sisteme sahip olurlar. Dolayısı ile o zamanlar merkezi bir sisteme sahip olan Amerika, dağıtık bir sisteme geçişin yollarını aramaktaydı. Böylece internet bir merkezde toplanmayan ve dağıtık bir sisteme sahip bir yapıdan oluşmaktadır.

İnternetin kendi içerisinde sunucu ve istemci olarak gelişen bir teknoloji olarak görülmesi mümkündür. İnternete bağlı olan her aygıt, internete hizmet etmektedir ve bu aygıtlar internet verilerinin tutulması veya dağıtılması için birer kaynak olarak görülebilir.

İnternet ise dinamik bir yapıya sahiptir ve verinin bazı kısımlarında hasar olması durumunda, başka yollar (ağlar) üzerinden verinin size ulaşabildiği bir yapıya sahiptir.

- ***Bir teknolojiye erişmek için, o teknolojiye ulaşmak için gereken bir kültürel seviyeye ulaşmanız da gerekmektedir.***
 - ***Teknolojiler, üzerlerinde kültür taşımaktadırlar.***

2.BÖLÜM: AĞ TEORİSİ

2.1 AĞ TEORİSİ – TEMEL KAVRAMLAR

2.2 AĞIN TAŞINMASI

2.3 VERİLERİN DÖNÜŞÜMÜ

2.4 VERİLERİN ULAŞTIRILMASI

2.5 AKTARIM HIZINI ETKİLEYEN FAKTÖRLER

2.6 GENİŞ BANDI ETKİLEYEN FAKTÖRLER

2.7 HESAPLAMALAR

2.1 AĞ TEORİSİ – TEMEL KAVRAMLAR

Ağ teorisi hakkında ileriki bölümlerde detayları ile ilgili bahsedilecek olup, bu bölümde temel kavramları üzerinde durulacaktır.

Ortam (Medium) olarak nitelendirdiğimiz kavram, bağlandığımız sistemi ifade etmektedir. Bağlanmış olduğumuz sistem üzerinde bulunan bilgisayarlar (veya benzeri aygıtlar) kendi arasında ikiye ayrılmaktadırlar. Bir kısmını veriyi alan (**istemci**) kısım oluştururken, diğer kısmını da veriyi sunan aygıtlar (**sunucu**) oluşturmaktadırlar.

Ortam kavramı internet bulunmadan önce de ortaya çıkan bir kavramdır. Örneğin internette önce, televizyonda ayarlanan kanaldan (sunucu) görüntünün bize ulaşmasında, televizyon bir istemci olarak düşünülebilir. Burada bize verinin iletim şekli sinyaldir. Bu sinyal aynı zamanda manyetik, optik, elektromanyetik ya da elektrik sinyalleri de olabilmektedirler.

Aslında bu kısımda iletilen her şey veridir. Teknolojinin geldiği bugün ki noktada, bildiğimiz her şeyi veriye çevirmeye çalışmaktayız. Veriyi çevirirken, modelleme yapabilmek için **bit** kavramından yararlanmaktayız.

Bit ile kast edilen ifade; 1 veya 0'lerden oluşan sayılardır. Bu sayılar **ikilik** (binary) **sistem** olarak da bilinmektedir. Bu sisteme göre veri, 1 veya 0 olabilmektedir. Böylece bitler birleşerek bizim için anlam ifade eden bir yapı oluşturmaktadır. Bu yapı aynı zamanda dijital sistemlerin de temelini oluşturmaktadır.

2.1.1 DİJİTAL SİSTEMLER

Sinyaller **analog** (sürekli sinyaller) ve **dijital** sistemler olarak ikiye ayrılmaktadırlar. Analog sinyaller; ses, elektrik sinyalleri veya radyo frekansı gibi sürekli gelen sinyallerdir. Bu sinyallerde veri bir değere sahiptir fakat bu değer herhangi bir değer olabilmektedir. Dijital sinyaller ise 1 ya da 0 değerini alabilen sinyallerdir. Böylece verinin bozulma ihtimali, analog sinyallere göre daha düşüktür.

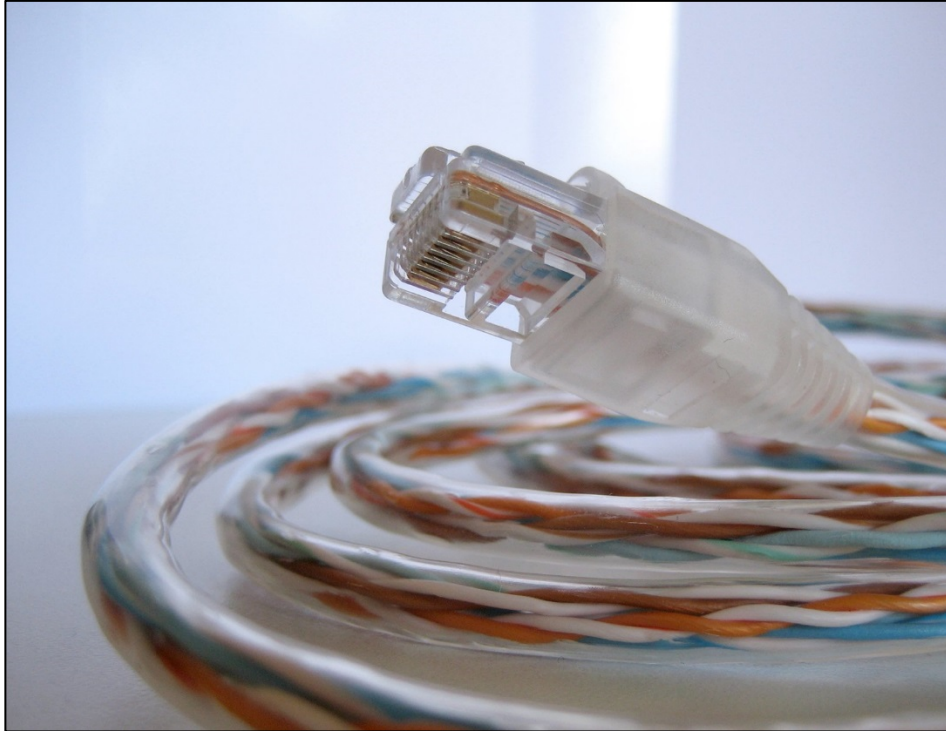
2.2 AĞ TAŞINMASI

Ağların (ya da diğer bir ifade ile bitlerin) taşındığı sistemlerde, veri öncelikle elektrik sinyalleri ile taşınabilmektedirler. Bu sinyallerin taşınması sağlayan **UTP**, **BNC** ve **Fiber Optik** kablolar bulunmaktadır.

UTP adını verdiğimiz kablolar ikili olarak birbirlerine dolanmış kablolardır ve birbirlerine dolanma sıklıkları farklıdır. Çünkü elektrik sinyallerini taşıdıkları için üzerlerinde belirli bir manyetik alan oluşturmaktadırlar. Oluşturulan manyetik alan başka bir kabloya değdiği zaman, orada da bir manyetik alan oluşturmaktadır. Dolayısıyla birbirlerinin manyetik alanlarını bozma riskini ortadan kaldırmak için, sarım sıklıkları farklı ayarlanmaktadır. Yani bir sinyalin daha hızlı gideceği yere ulaştırılması ve diğerinin ondan daha yavaş olarak oraya ulaştırılması sağlanmaktadır.

2.2.1 UTP

UTP adını verdiğimiz kablolar ikili olarak birbirlerine dolanmış kablo tipleridir ve birbirlerine dolanma sıklıkları farklıdır. Çünkü elektrik sinyallerini taşıdıkları için üzerlerinde belirli bir manyetik alan oluşturmaktadırlar. Oluşturulan manyetik alan başka bir kabloya değdiği zaman, aynı zamanda bir manyetik alan daha oluşturmaktadır. Dolayısıyla birbirlerinin manyetik alanlarını bozma riskini ortadan kaldırmak için, sarım sıklıkları farklı ayarlanmaktadır. Özetle bir sinyalin daha hızlı gideceği yere ulaştırılması ve diğerinin ondan daha yavaş olarak ulaştırılması sağlanmaktadır.



Resim 2.2.1.1

2.2.2 BNC

Diğer bir sinyallerin taşınması sağlayan kablo türü de BNC kablolarıdır. Bu tip kabloların yapısı katmanlıdır. En içte bakır bir tel ve üzerlerinde metal yapı bulunmaktadır. Bu metal yapının amacı, olası sinyallerin dinlenmesi vb. gibi güvenlik sorunlarının önüne geçerek, korunmasını sağlamaktır.



Resim 2.2.2.1

2.2.3 FİBER OPTİK

Günümüzde elektrik sinyallerinin taşınmasına bakıldığı zaman, fiber optik kablolar bulunmaktadır. Bu tip kablolar ışığı kullanarak yapılandırılmışlardır ve artık günümüzde birden fazla veriyi aynı zamanda taşınması da sağlanarak geliştirilmişlerdir.



Resim 2.2.3.1

2.3 VERİLERİN DÖNÜŞÜMÜ

İkilik sistem (Binary), veri için açık/kapalı, evet/hayır, 1/0, yüksek/düşük gibi cevapların elde edilmesini sağlamaktadır.

Aynı zamanda, ikilik sistem ile birlikte byte kavramını da elde etmekteyiz. **Byte**, 8 tane bit'in bir araya gelerek oluşturduğu değerdir. Byte kavramı, bit'e göre daha anlamlı bir yapıya sahiptir. Bit'lerde anlamlı bir parçayı oluşturmaktadırlar fakat 1 veya 0 olarak ifade edilen bir veri, tek başına iken anlamlı değildir. Dolayısıyla byte kavramı, 8 tane bitten oluştuğu için 256 (Bir bit 2 değer alabildiği ve 8 bitten oluştuğu için: $2^8 = 256$) tane değer kodlayabilmektedir.

Bit (b) = Binary Digit

Byte kavramı bizim için 0'dan 255'e kadar numaraları değer ifade edebileceği gibi bir harfi de ifade edebilir. Örneğin **ASCII tablosu**, ikilik sistemi kullanarak oluşturulmuş bir tablodur. Böylece bilgisayarın çalışma mantığında her bir sembolün (harf, sayı vb.), Ascii tablosunda bir karşılığı bulunmaktadır.

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
\$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
;	59	0073	0x3b	[91	0133	0x5b	{	123	0173	0x7b
<	60	0074	0x3c	\	92	0134	0x5c		124	0174	0x7c
=	61	0075	0x3d]	93	0135	0x5d	}	125	0175	0x7d
>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
?	63	0077	0x3f	_	95	0137	0x5f				

Resim 2.3.1

Byte deęerleri arasında dnüşürme yapılacağı zaman, ikilik sistemi kullandığımız için her zaman deęerin $2^{10} = 1024$ ile çarpımı alınmaktadır. Dolayısıyla ařaęıda verilen byte deęerinin basamaklarını kendi aralarında dnüşürmek istediğimiz zaman 2^{10} ile çarparak ya da bölerek elde edilebilmekteyiz.

BELLEK ÖLÇÜ BİRİMLERİ		
BİRİM	KISALTMA	KAPASİTE
Byte	b	8 bit
Kilobyte	kb	1024 byte
Megabyte	mb	1024 kilobyte
Gigabyte	gb	1024 megabyte
Terabyte	tb	1024 gigabyte
Petabyte	pb	1024 terabyte
Exabyte	eb	1024 exabyte

2.4 VERİNİN ULAŞTIRILMASI

Herhangi bir mesajın ya da resmin bir yere ulaşmasını istediğimiz zaman, elimizdeki veri, kullanılan uygulama tarafından ilk olarak 1 ve 0'lardan oluşan değere dönüştürülür ve daha sonra o hat üzerinden karşı tarafa iletilir. Örneğin Whatsapp üzerinden gönderiyorsanız Whatsapp, Twitter üzerinden gönderiyorsanız da Twitter bu hizmeti sağlamaktadır.

2.5 AKTARIM HIZINI ETKİLEYEN FAKTÖRLER

Verilerin iletim hızlarına bakıldığı zaman **latency** adını verdiğimiz bir kavram ön plana çıkmaktadır. Latency, Türkçe de bir işin yapılması için geçecek olan zamanı ifade etmektedir. Örneğin bir filmi, müziği ya da resmi indirmek için beklediğiniz zaman, latency kavramı ile açıklanmaktadır.

Diğer bir aktarım hızını etkileyen faktör ise sinyal hızıdır (**Signal rate**) ve bir sinyalin karşı tarafa aktarılması için geçecek olan zamanı ifade etmektedir.

Son faktör ise **Transfer Rate** veya **Bandwidht** olarak karşımıza çıkan **bant genişliğidir**. Bant genişliği verinin ne kadar hızlı indirilebildiğini ifade etmektedir. Yani örneğin 4 mbps demek, sizin bilgisayarınızın saniyede 4 mega bit indirdiğini ifade etmektedir

Aşağıdaki resimde görmüş olduğunuz arabalar veriyi temsil etmektedirler.



Resim 2.5.1

NOT: Dikkat edilmesi gereken kısım, aktarım hızının ile sinyal hızının aynı şey olmadığıdır. Aksine aktarım hızı, sinyal hızına da bağlı olan bir durumdur. Aynı anda birden fazla sinyal gönderilmesi (iletim kapasitesi) ve sinyal hızları, verinin aktarım hızını etkilemektedir.

Aşağıdaki resimde iletim kapasitesi ile sinyal hızının arasındaki bağlantı açıklanmıştır. Alçak sinyal hızı ile düşük iletim kapasitesi, zayıf aktarım hızı ortaya çıkmaktadır. Bu durumun tam tersinde ise, yani yüksek sinyal hızı ve yüksek iletim kapasitesinde, hızlı aktarım hızı ortaya çıkmaktadır.

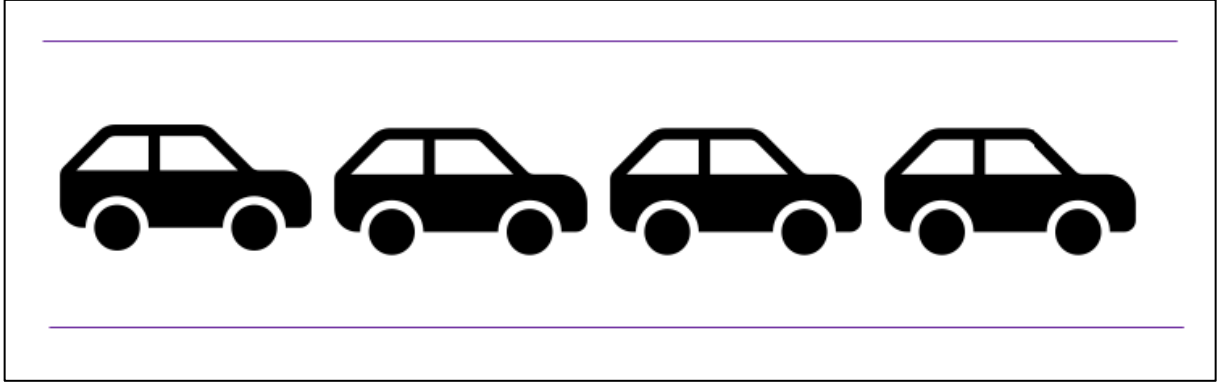
Diğer durumlarda alçak sinyal hızı ve yüksek iletim kapasitesinde, aktarım hızında sıkışıklık meydana gelmektedir. Yüksek sinyal hızı ve düşük aktarım hızında ise aktarım hızının seyrekleştiği görülmüştür.

SİNYAL HIZI			
İLETİM KAPASİTESİ	ALÇAK	ALÇAK ZAYIF	YÜKSEK SEYREK
	YÜKSEK	SIKIŞIK	HIZLI

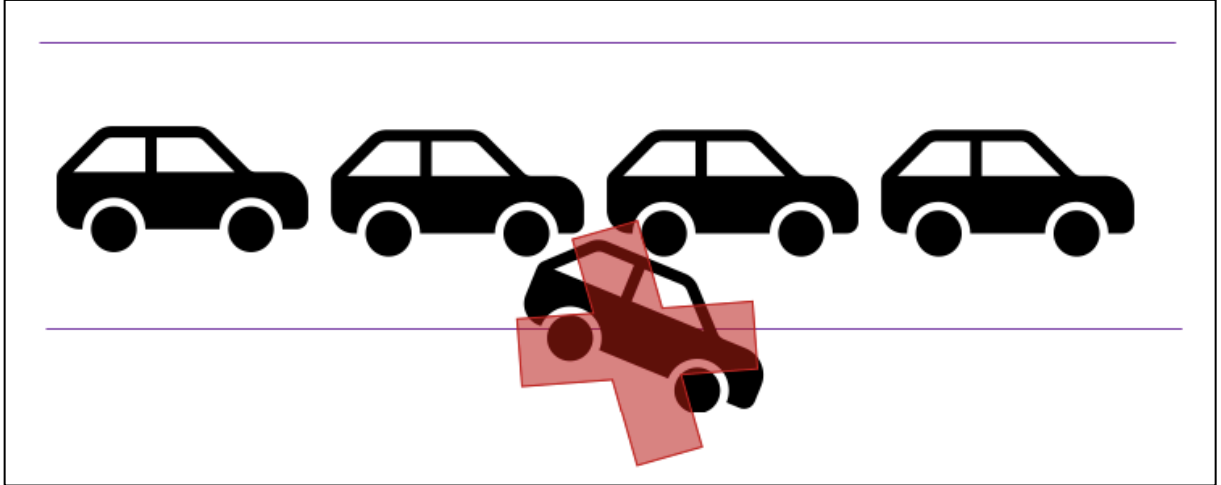
Resim 2.5.2

2.6 GENİŞ BANDI ETKİLEYEN FAKTÖRLER

Geniş bandı etkileyen ilk faktör, **sinyalin miktarıdır**. Geniş bandı artırabilmek için tek bir yoldan sayıca fazla sinyal göndermemiz mümkündür (aynı şekilde azaltmak için de sinyal miktarının düşürülmesi gerekmektedir). Fakat unutulmamalıdır ki bizim gönderebildiğimiz sinyalin belirli bir sınırı bulunmaktadır.

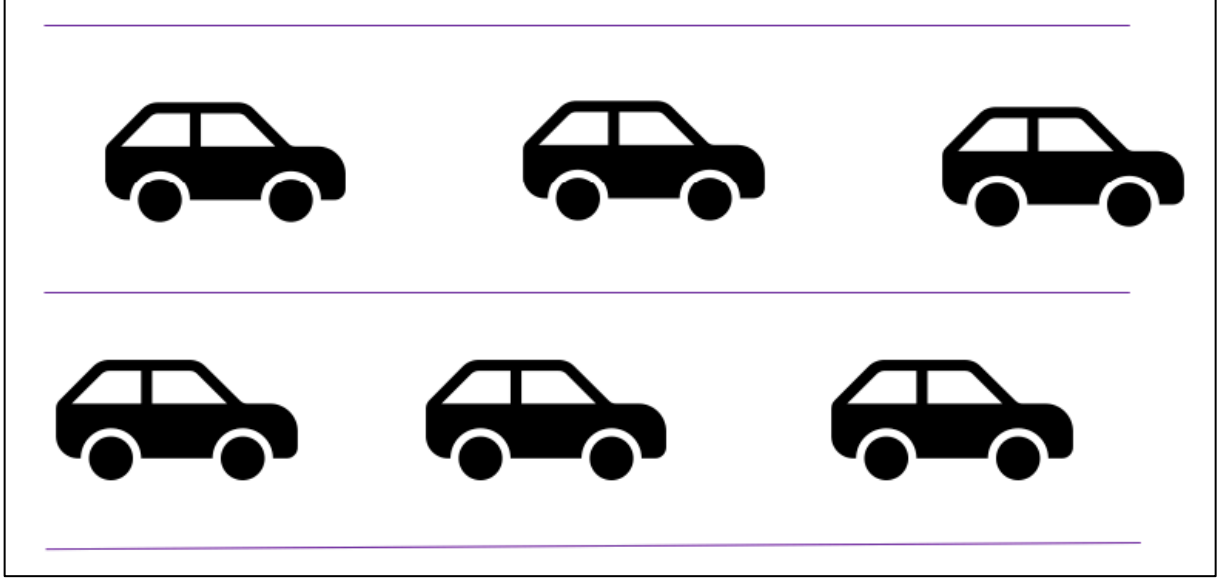


Resim 2.6.1



Resim 2.6.2

Geniř bandı etkileyen bir dięer faktör ise **yol sayısıdır**. Yol sayısını arttırmak, geniř bandın da artmasına sebep olacaktır. Fakat bu yol dięer çözümlere göre daha maliyetli olabilmektedir. Çünkü netice itibari ile yeni bir sinyal hattı döřemeniz gerekmektedir.



Resim 2.6.1

2.7 HESAPLAMALAR

2.7.1 VERİNİN İLETİM SÜRESİNİN HESAPLANMASI

Bir verinin başka bir yere ulaşırken harcadığı zamanı hesaplamak için, verinin boyutunu, iletim hızına bölmeniz yeterlidir. Bahsi geçen ifadenin formüle edilmiş hali aşağıda verilmiştir:

$$\text{VERİNİN İLETİM SÜRESİ}(s) = \frac{\text{DOSYANIN BOYUTU}(B)}{\text{İLETİM HIZI}\left(\frac{B}{s}\right)}$$

2.7.1 VERİNİN İLETİM HIZININ HESAPLANMASI

Bir önceki adımda verilen formülden yola çıkarak, iletim hızını hesaplamak için, verinin boyutunu, verinin iletim süresine bölünmesi gerekmektedir. İletim hızının formüle edilmiş hali aşağıda verilmiştir:

$$\text{İLETİM HIZI}\left(\frac{B}{s}\right) = \frac{\text{DOSYANIN BOYUTU}(B)}{\text{VERİNİN İLETİM SÜRESİ}(s)}$$

3.BÖLÜM: İNTERNET PROTOKOLÜ,

DNS VE ALAN İSİMLERİ

3.1 İNTERNET NEDİR?

3.2 İNTERNET DONANIMI

3.3 İNTERNET YAZILIMI

3.4 PAKET ANAHTARLAMA

3.5 İNTERNET PROTOKOLÜ

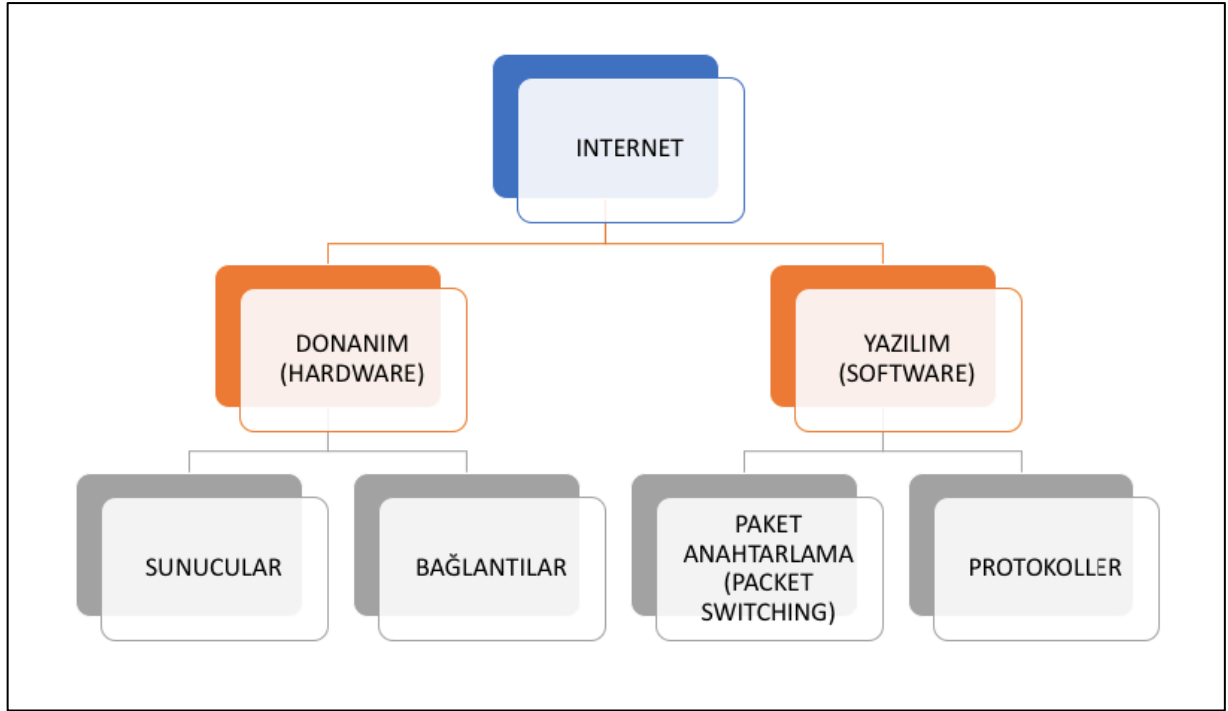
3.6 DOMAIN İSİMLERİ

3.7 DNS (DIRECT NAME SERVER)

3.8 İNTERNET POLİTİKALARI

3.1 İNTERNET NEDİR?

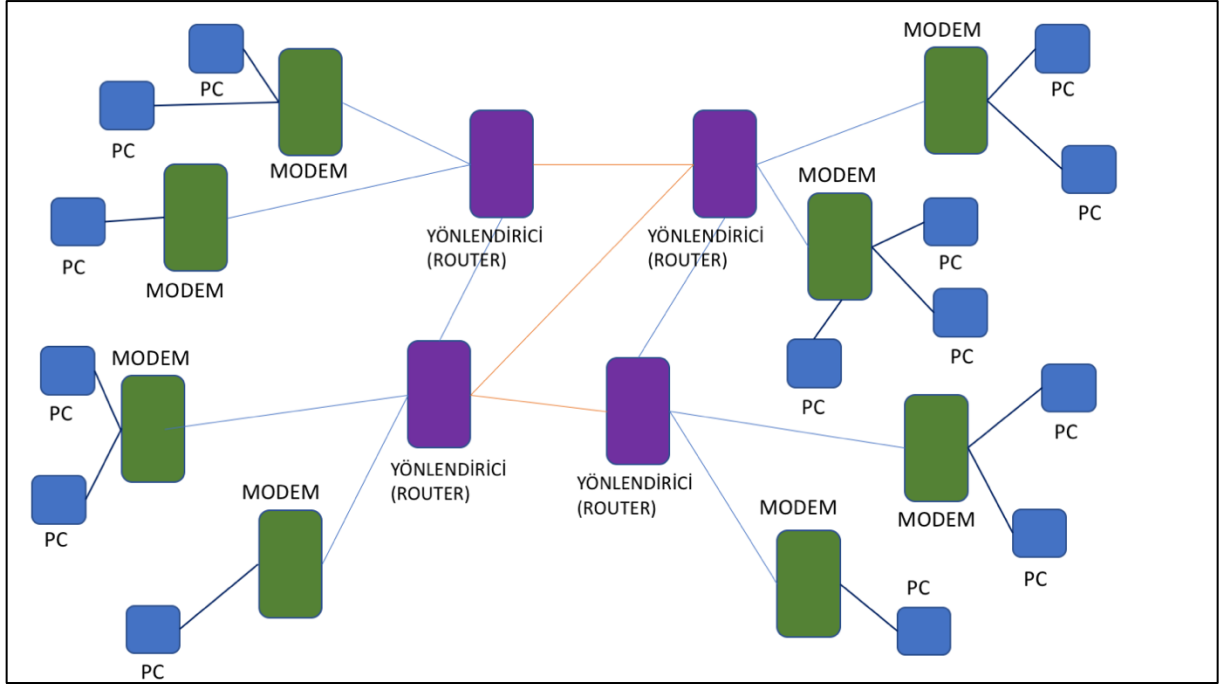
Resim 3.1.1 de görüldüğü üzere interneti **donanım** ve **yazılım** olarak ikiye ayırmak mümkündür. Donanım kısmında iletişimin doğru kurulabilmesi için bir sürü cihaz (sunucu ya da server, istemci vb.) ve bağlantılar bulunurken, yazılım kısmı ise protokollerden ve paketlerden oluşmaktadır. Kitabın bu bölümünde her iki alan da detayları ile incelenecektir.



Resim 3.1.1

3.2 İNTERNET DONANIMI

İnternet ağlarının yapısı incelendiği zaman, en son uçta bizim istemci bilgisayarlarımız (vb. bağlantı cihazlarımız) bulunmaktadır. Onların haricinde ana akış hattı üzerinde **sunucu** (server) ve **yönlendiriciler** (router) bulunmaktadır. Fakat bu ağlardan herhangi bir bağın hasar görmesi durumunda, diğer bir ağ üzerinden internetin sağlanması mümkün olmaktadır.



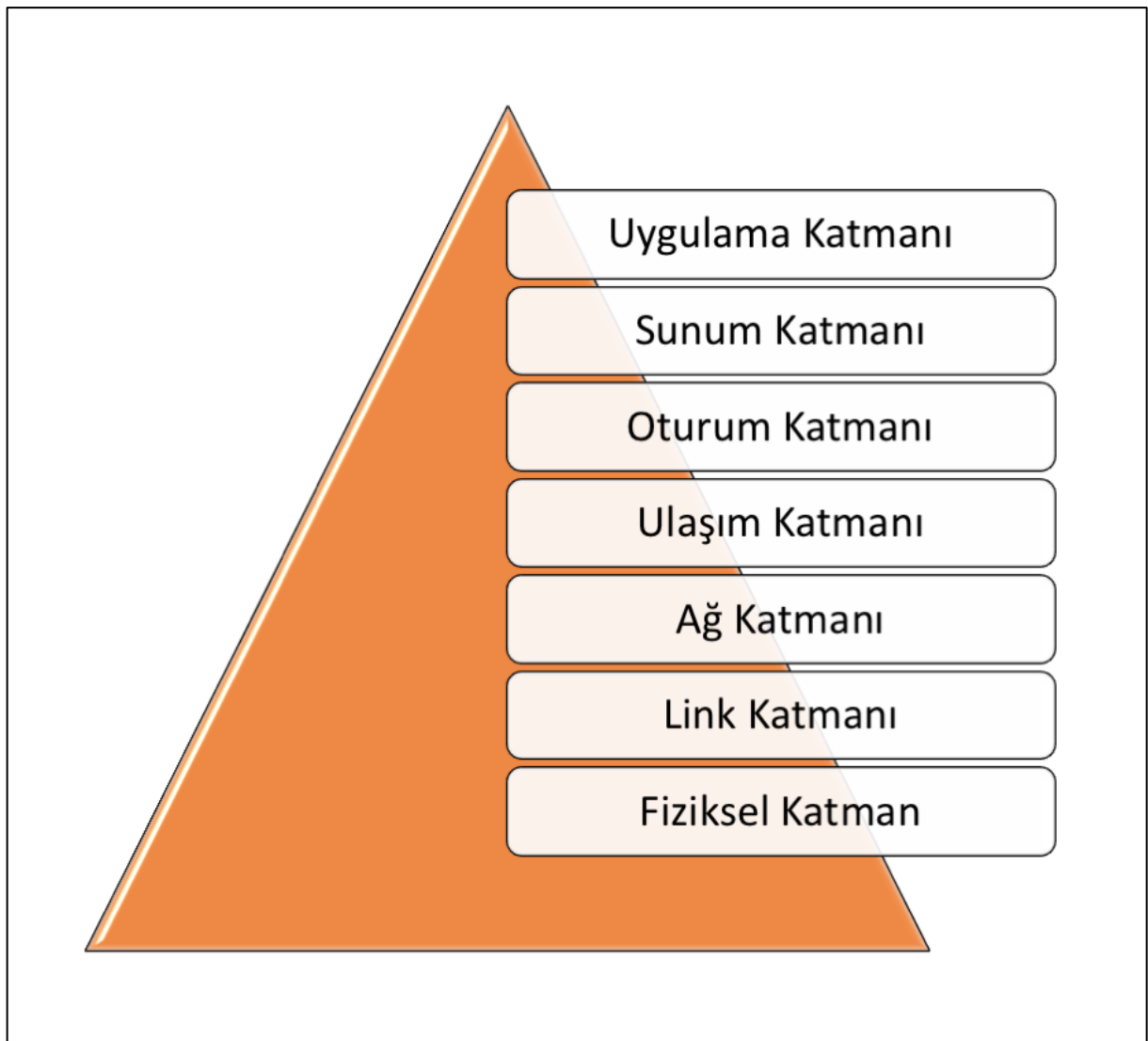
Resim 3.2.1

3.3 İNTERNET YAZILIMI

İnternet katmanlardan oluşmaktadır ve bu katmanların en alt kısmında **fiziksel katman** bulunmaktadır. Fiziksel katmanda, geçen bölümlerde de bahsettiğimiz elektrik, elektromanyetik vb. sinyallerden oluşan yapılar yer almaktadırlar.

Fiziksel katmanın hemen üst kısmında **link katmanı** bulunmaktadır. Link katmanının amacı iki bilgisayar arasında bağlantının hattını sağlamaktır.

Link katmanının hemen üstünde ise **ağ (network) katmanı** yer almaktadır. IP (Internet Protocol), ağ katmanı üzerinde bulunmaktadır.

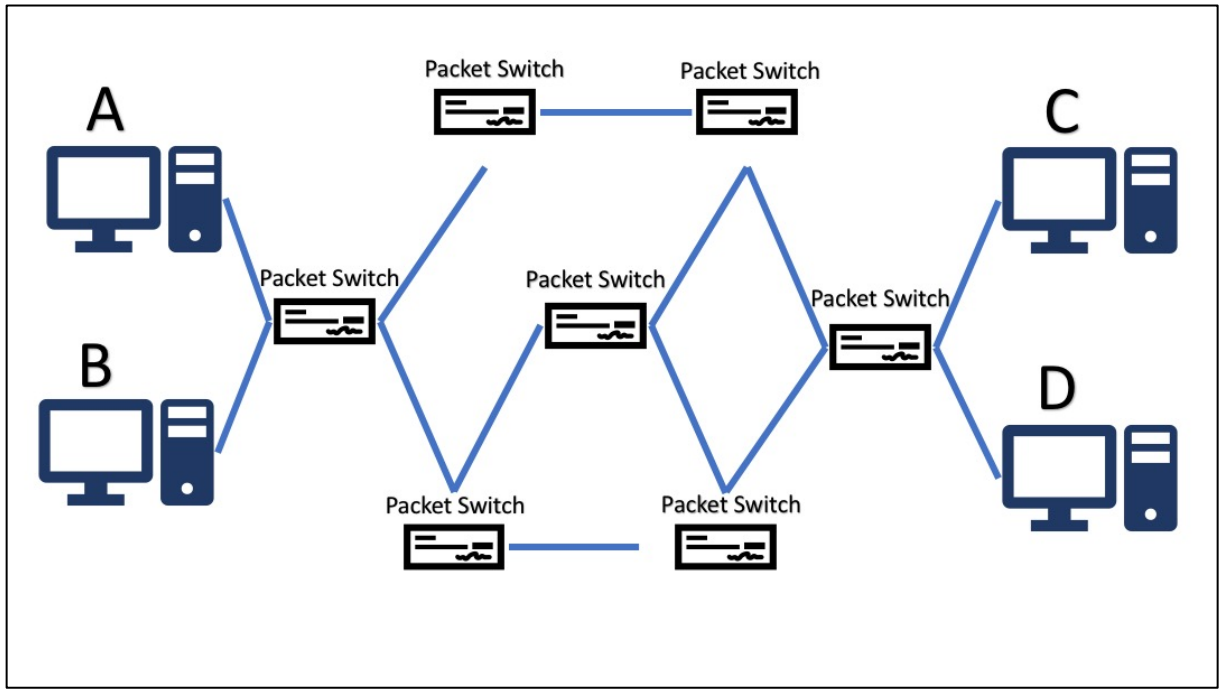


Resim 3.3.1

3.4 PAKET ANAHTARLAMA

3.4.1 PAKET ANAHTARLAMANNIN TARİHİ

Paket anahtarlama işleminin tarihine bakıldığında zaman, geçmişte bu işlem için kullanılan ilk yöntemlerden biri, kişiye özel kanal tahsis edilmesiydi. Kanalın tahsis edilmesi ile birlikte, yetkili kişi müşterinin kablosunu bağlantı kurmak istediği diğer kişinin kablosuna bağlayarak iletişimi sağlamış oluyordu. Fakat bu yöntem aynı zamanda maliyetli ve uğraştırıcı olduğu için insanlar tarafından çok da tercih edilmedi. Fakat aynı yöntem sunucular için tercih edilebilen bir yöntemdi.

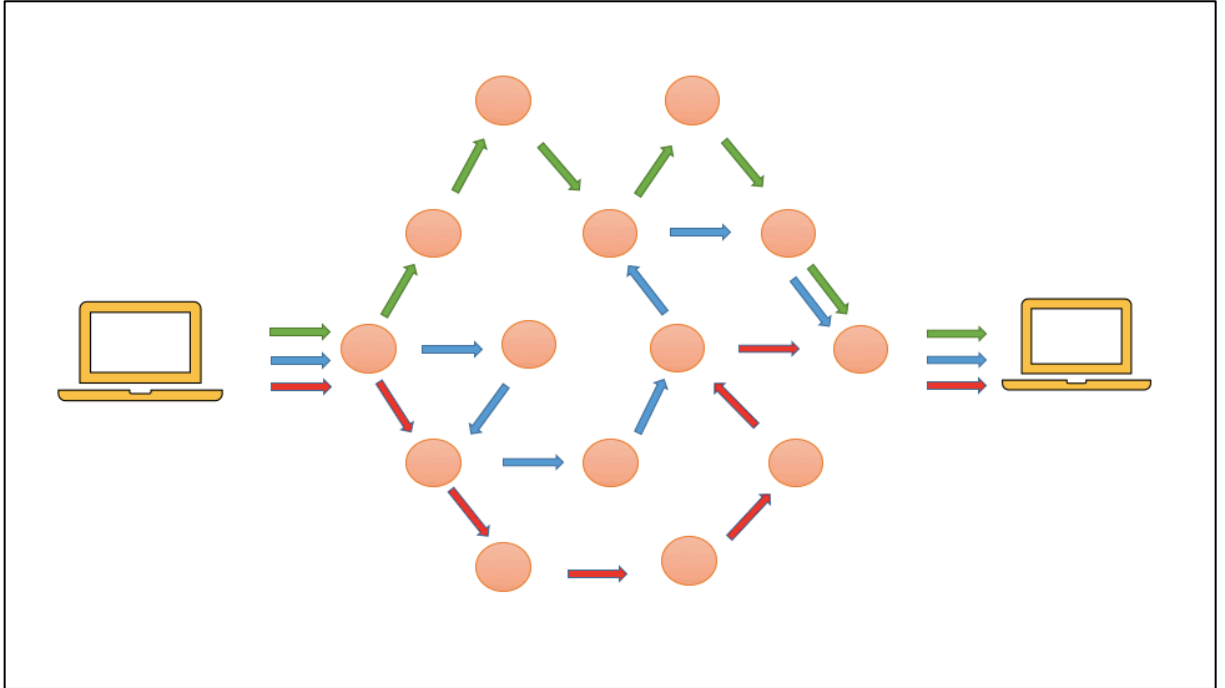


Resim 3.4.1.1

3.4.2 PAKET ANAHTARLAMA NEDİR?

İki bağlantı noktasının, olası tüm hatlar ve bağlantı noktaları üzerinden birbirine bağlanmayı denemesine paket anahtarlama adı verilmektedir. Paketler bu yollar üzerinden ulaşabilecek en uygun yolu yönlendiriciler sayesinde tercih etmektedirler.

NOT: Resim 3.4.2.1 de görüldüğü üzere, turuncu olarak simgelenen yönlendiriciler, ağ noktaları üzerinde bir IP'nin bir sonraki noktasını belirlemektedirler. Örneğin bir IP yola çıktığı zaman hangi yollardan geçerek hedefe varacağını kesin olarak bilemez. Dolayısıyla her gittiği diğer bağlantı noktasındaki yönlendirici, onun bir sonraki yönünü belirler ve böylelikle gideceği yere ulaşımı sağlar.



Resim 3.4.2.1

3.5 İNTERNET PROTOKOLÜ

IP adresi internet üzerinde verilerin size ulaşmasını sağlamaktadır. İnternet protokolü 134.168.11.1 gibi 4 haneli sayılardan oluşmaktadır ve aralarına nokta konularak ayrılan her sayı 0'dan 255'e kadar (256 tane) değer alabilmektedir. Daha önceki bölümlerde belirttiğimiz üzere, 256'ya kadar değer alabilen her küme 1 byte'lık (8 bitlik) yer kaplamaktadır. Dolayısıyla IP numarası ayrı ayrı 4 haneye sahip olduğu için 32 bitlik veri bulundurmaktadır.

Aşağıda örnek olarak verdiğimiz yönlendirme tablosunda IP adreslerinin, bir yere ulaşmak için çıktıklarında, aslında tam olarak rotasının belli olmadığını görüyoruz.

YÖNLENDİRİCİ TABLOSU	ADRES	HEDEF
	131.229.X.X	131.229.14.11
	128.118.X.X	128.118.2.249
	148.85.X.X	148.85.34.9

Resim 3.5.1

Bunun sebebi, IP adresleri verileri ulaştırırken direk olarak oraya gitmemeleridir. Hedefe varmadan önce farklı IP adreslerine uğrar ve oradaki yönlendiricilerden gideceği adresi alırlar.

NOT: Protokol, bir ağ üzerindeki bilgisayarlar arasındaki iletişimi yöneten kurallar kümesidir.

3.5.1 IP ADRESİ NASIL ÖĞRENİLİR?

IP Adresinizi kolaylıkla bilgisayarınızda bulunan terminal üzerinden öğrenebilirsiniz. Bunun için öncelikle MAC ve Linux terminallerinde neler yapmanız gerektiğini, daha sonra da bu kodların Windows ortamındaki karşılıklarından bahsedeceğiz.

NOT: Eğer terminal kullanacaksanız MAC ve Linux da yazılan kodlar birbiri ile aynıdır.

MAC veya Linux terminali açıldıktan sonra **ifconfig** komutunu yazmanız kendi bilgisayarınızın IP adresini öğrenmeniz için yeterlidir. **Loc** olarak geçen ilk kısımda yerel (local) IP adresiniz **inet** olarak belirtilmiştir. Kendi bilgisayarınızın özel IP adresi ise ikinci kısım da (yani **en0**), yine **inet** karşılığında belirtilmiştir. Herhangi birinin bu IP adresini kullanarak sizin bilgisayarınıza bağlanması mümkündür.

Resim 3.5.1.1 de terminalde ifconfig komutunun çalıştırılmış hali mevcuttur:

```
Last login: Sun Aug 5 12:25:53 on ttys000
[Bilgisayars-MacBook-Air:~ bilgisayaravramlari$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM, TXCSUM, TXSTATUS, SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHC20: flags=0<> mtu 0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 1c:36:bb:29:17:d6
    inet6 fe80::4f4:dc9411:754b%en0 prefixlen 64 secured scopeid 0x5
    inet 10.10.0.61 netmask 0xffff0000 broadcast 10.10.255.255
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
    ether 0e:36:bb:29:17:d6
    media: autoselect
    status: inactive
awd10: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
    ether b6:8b:39:fe:55:af
    inet6 fe80::b48b:39ff:fefe:55af%awd10 prefixlen 64 scopeid 0x7
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=60<TS04,TS06>
    ether 9a:00:13:f0:ef:60
    media: autoselect <full-duplex>
    status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=63<RXCSUM, TXCSUM, TS04, TS06>
    ether 9a:00:13:f0:ef:60
    Configuration:
        id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
        maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
        root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
        ipfilter disabled flags 0x2
    member: en1 flags=3<LEARNING,DISCOVER>
        ifmaxaddr 0 port 8 priority 0 path cost 0
    nd6 options=201<PERFORMNUD,DAD>
    media: <unknown type>
    status: inactive
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
    inet6 fe80::8ecb:8d7b:905:dc4e%utun0 prefixlen 64 scopeid 0xa
    nd6 options=201<PERFORMNUD,DAD>
Bilgisayars-MacBook-Air:~ bilgisayaravramlari$
```

Resim 3.5.1.1

IP adresinin hedef IP'yi nasıl bulduğunu daha iyi anlayabilmek için bir komut daha çalıştıralım. Bu komut **tracert** komutudur ve asıl IP adresinin bizim bilgisayarımıza ulaşana kadar nasıl bir yol izlediğini gösterir.

Terminalde kodumuzu çalıştırdığımız zaman, sizden web sitesine bağlanmak için adres girmenizi isteyecektir. Örnek olarak Resim 3.5.1.2 de www.bilgisayarkavramlari.com sitesine ulaşmaya kadar IP adresimizin geçtiği yollar gözükmetedir (ilk satırlardan sonra IP adreslerinin daha karmaşık bir yapı almasının sebebi, dünya üzerindeki diğer cihazlara bağlanmasından kaynaklanmaktadır.).

```
Bilgisayars-MacBook-Air:~ bilgisayar_kavramlari$ tracert www.bilgisayarkavramlari.com
tracert to bilgisayar_kavramlari.com (184.168.221.20), 64 hops max, 52 byte packets
 1  hotspot.akinogrenciyurdu.com (10.10.10.1)  1.719 ms  1.147 ms  1.346 ms
 2  192.168.0.1 (192.168.0.1)  1.847 ms  5.308 ms  5.115 ms
 3  * * *
 4  172.25.7.1 (172.25.7.1)  10.740 ms  11.017 ms  12.612 ms
 5  195.175.76.165.static.turktelekom.com.tr (195.175.76.165)  9.695 ms  13.286 ms  9.559 ms
 6  07-antalya-t3-5---07-calli-t3-r.statik.turktelekom.com.tr (81.212.216.69)  9.863 ms  10.098 ms  11.276 ms
 7  07-antalya-xrs-t2-2---07-antalya-t3-5.statik.turktelekom.com.tr (212.156.109.37)  19.345 ms  17.725 ms  11.310 ms
 8  06-ulus-xrs-t2-2---07-antalya-xrs-t2-2.statik.turktelekom.com.tr (81.212.220.3)  18.811 ms  25.579 ms  26.289 ms
 9  302-ams-col-1---06-ebgp-ulus1-k.statik.turktelekom.com.tr (212.156.102.180)  66.344 ms  66.002 ms  63.746 ms
10  212.119.24.245 (212.119.24.245)  79.341 ms  62.127 ms  61.828 ms
11  ae-6.r24.amstnl02.nl.bb.gin.ntt.net (129.250.3.225)  66.686 ms
    ae-8.r25.amstnl02.nl.bb.gin.ntt.net (129.250.3.229)  69.638 ms  76.161 ms
12  ae-1.a01.amstnl02.nl.bb.gin.ntt.net (129.250.4.173)  173.433 ms  447.728 ms  175.076 ms
13  * ae7.edge6.amsterdam.level3.net (4.68.63.217)  87.843 ms  63.366 ms
14  * * *
15  4.28.83.74 (4.28.83.74)  276.712 ms  280.612 ms  306.427 ms
16  184.168.9.0 (184.168.9.0)  298.701 ms  335.922 ms  306.850 ms
17  * * *
```

Resim 3.5.1.2

Deđinilmesi gereken bir diđer komutumuz **Ping**, herhangi bir siteye bađlanmak ve o sitenin bundan haberi olması anlamını tařıtmaktadır. Terminalde alıřtırıldıđı zaman iki cihazın hangi paketlerden ne kadar zamanda getiđini gstermektedir.

```
Last login: Sun Aug 5 12:26:33 on ttys000
Bilgisayars-MacBook-Air:~ bilgisayar kavramlari$ ping www.bilgisayarkavramlari.com
PING bilgisayar kavramlari.com (184.168.221.20): 56 data bytes
64 bytes from 184.168.221.20: icmp_seq=0 ttl=44 time=202.002 ms
64 bytes from 184.168.221.20: icmp_seq=1 ttl=44 time=202.594 ms
64 bytes from 184.168.221.20: icmp_seq=2 ttl=44 time=303.322 ms
64 bytes from 184.168.221.20: icmp_seq=3 ttl=44 time=207.204 ms
64 bytes from 184.168.221.20: icmp_seq=4 ttl=44 time=203.020 ms
64 bytes from 184.168.221.20: icmp_seq=5 ttl=44 time=231.575 ms
64 bytes from 184.168.221.20: icmp_seq=6 ttl=44 time=206.580 ms
64 bytes from 184.168.221.20: icmp_seq=7 ttl=44 time=290.987 ms
64 bytes from 184.168.221.20: icmp_seq=8 ttl=44 time=211.390 ms
64 bytes from 184.168.221.20: icmp_seq=9 ttl=44 time=201.682 ms
64 bytes from 184.168.221.20: icmp_seq=10 ttl=44 time=365.071 ms
64 bytes from 184.168.221.20: icmp_seq=11 ttl=44 time=335.356 ms
64 bytes from 184.168.221.20: icmp_seq=12 ttl=44 time=317.264 ms
64 bytes from 184.168.221.20: icmp_seq=13 ttl=44 time=200.813 ms
64 bytes from 184.168.221.20: icmp_seq=14 ttl=44 time=205.362 ms
64 bytes from 184.168.221.20: icmp_seq=15 ttl=44 time=203.938 ms
64 bytes from 184.168.221.20: icmp_seq=16 ttl=44 time=208.630 ms
64 bytes from 184.168.221.20: icmp_seq=17 ttl=44 time=215.331 ms
64 bytes from 184.168.221.20: icmp_seq=18 ttl=44 time=204.691 ms
64 bytes from 184.168.221.20: icmp_seq=19 ttl=44 time=201.148 ms
64 bytes from 184.168.221.20: icmp_seq=20 ttl=44 time=256.997 ms
█
```

Resim 3.5.1.3

Aynı komutları Windows terminalinde alıřtırmak istersek, ifconfig yerine **ipconfig** ve traceroute yerine de **tracert** yazmanız yeterlidir. Ping komutu Windows terminalinde de ping olarak iřlev grmektedir.

NOT: ‘traceroute’ komutunun iřlevini yerine getiren siteler mevcuttur. Arama motorunuza ‘**online tracerouter**’ ya da ‘**visual traceroute**’ yazarsanız bu gibi sitelere ulařabilirsiniz.

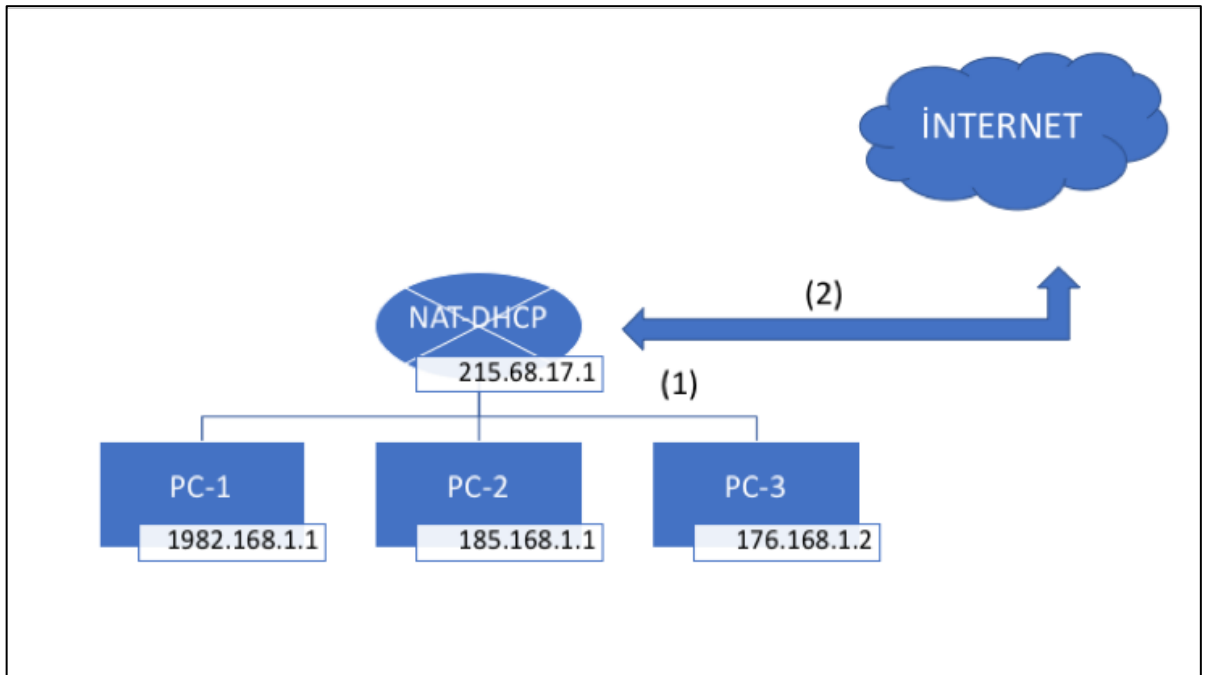
3.5.2 IP ADRESLERİNİN ULAŞILABİLİRLİĞİ

IP adresleri günümüzde birçok soruna sahip olan bir konudur. Bunun nedeni her biri 32 bitlik IP adreslerinin birleşimi ile dünya üzerinde 4 milyon (2^{32}) civarında IP adresi bulunmasıdır.

Aynı zamanda internete bağlı cihaz sayısına baktığımız zaman, dünya üzerinde 8 milyon civarında insan olduğu varsayılırsa ve herkesin sadece tek bir aygıtı olduğu bile göz önüne alınsa IP adreslerinin sayısını geçmektedir. Kaldı ki herkesin bilgisayar hariç ayrıca tableti ve telefonu olma ihtimali çok yüksektir. Bunların dışında, kimseye bağımlı olmayan sunucular bulunmaktadır. Dolayısı ile 4 milyona yakın IP adresinin olması ve 8 milyona yakın cihazın internete bağlanmak istemesi büyük bir sorun teşkil etmektedir.

Bu kısımda öncelikle bu sorunun **IPv4 (IP Version 4)** üzerinde nasıl çözüme kavuşturulduğundan bahsedilecektir. Daha sonra da niye **IPv6 (IP Version 6)** sisteminde geçildiğine değinilecektir.

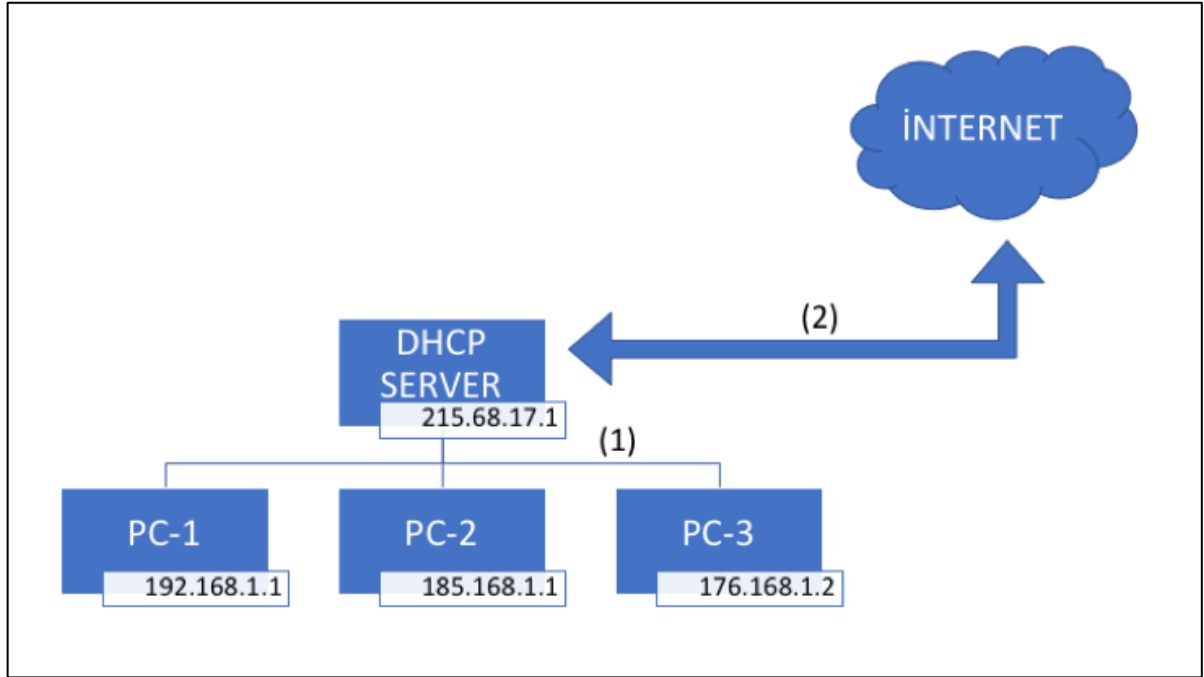
IPv4 üzerinde bu sorunun çözümünde **DHCP (Dynamic Host Configuration Protocol)** sunucu cihazları yer almaktadır. DHCP, aldığı IP adresine kendi içerisinde ayrıca IP adresi vermektedir. Yani eğer iç network'e bağlı bir gezinme işlemi gerçekleştiriyorsanız DHCP'nin atadığı bir IP ile (1), dış network'e bağlı bir gezinme işlemi gerçekleştiriyorsanız DHCP'nin kendi IP adresi ile (2) internete bağlanmaktasınız.



Resim 3.5.2.1

Dış network üzerinden size gelen veri paketleri DHCP'nin kendi IP adresine gelmektedirler. DHCP üzerinden internete bağlanan cihazlar arasından, veri paketinizi doğru bir şekilde sizin IP adresinize teslim edebilmesi için ise **NAT (Network Translation System)** teknolojisi kullanılmaktadır.

NAT cihazı kendi üzerinde internete bağlanan cihazların, hangi porttan bağlandıklarının bilgisini tutmaktadır. Dolayısı ile sizin IP adresinize gelen veri paketini size ait olan porta göndermektedir.



Resim 3.5.2.2

IPv4 ile getirilen çözüm göz önüne alındığında, ilk adımda 8 milyona yakın cihaz kendi IP adresi ile internete bağlanacak iken, şuan 4 milyona yakın erişim noktası bulundurmaktayız. Dolayısı ile kendi içerisinde IP adresi atamak bu büyük sorunun ortadan kaldırılmasına yardımcı olmuştur. Fakat 4 milyon tane erişim noktasına sahip olmamızda bir zaman sonra bir sıkıntı oluşturmuştur. Dolayısı ile IPv6 geliştirilmiştir. IPv4'te 32 bitten yani başka bir ifadeyle sekiz bitlik 4 rakamdan oluşmaktayken, IPv6, 128 bitlik 6 haneden oluşmaktadır.

3.5.3 IP ADRESLERİNİN ÖZELLİKLERİ

IP adreslerinde **başlık (header)** ve **veri (data)** kısmı bulunmaktadır. Veri kısmı kullanılan verinin boyutuna göre uzay alanda değişebilmekle birlikte, başlık kısmı sabit kalmaktadır.

Başlık kısmı 20 byte yan 160 bitlik alana sahiptir ve 40 onaltılık düzen (hexadecimal digit) içermektedir. Bu kısmın içerisinde IP adresinin hangi versiyon olduğu, servis boyutu, paketin veri kısmının boyutu, IP'nin nereden geldiği, gönderilen verinin hasarlı olup olmadığı ya da IP'nin ne kadar geçerli olacağı gibi bilgileri tutmaktadır.



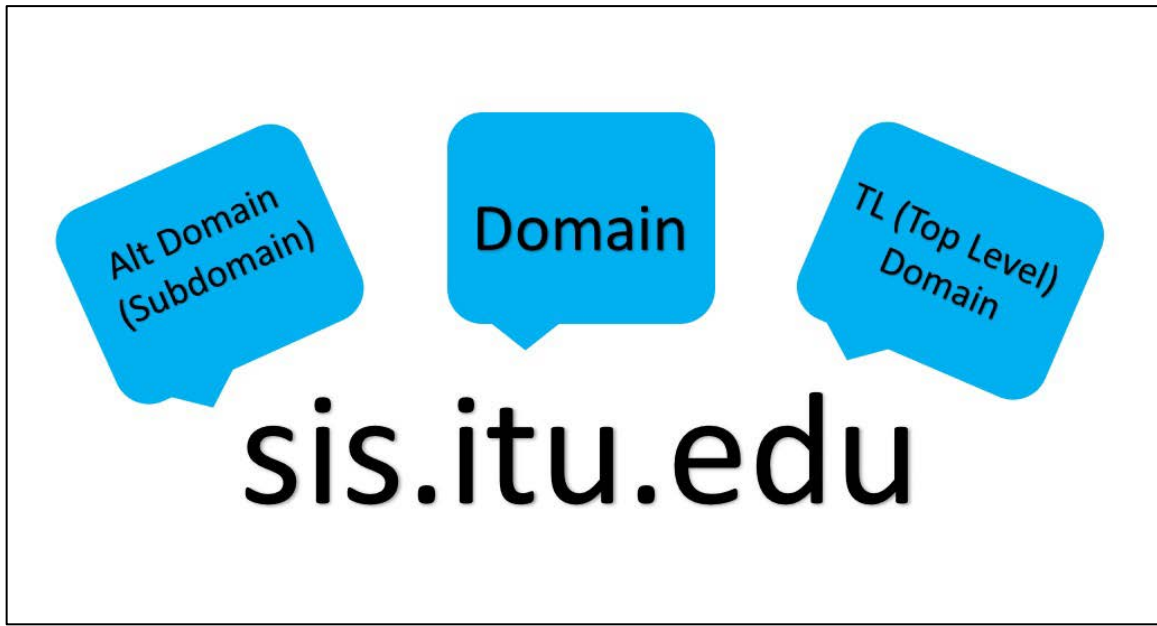
Resim 3.5.3.1

3.6 DOMAIN İSİMLERİ

İnternet adreslerinde orta kısımda yer alan kısım asıl **domain** ismini içermektedir. Bu isimleri belirli bir ücret karşılığında ve daha önce kullanılmamış olması koşulu ile siz belirleyebilmektesiniz.

Alt domain olarak belirtilen ilk kısmı yine sizin belirleyebilmeniz ile birlikte, size hangi adres uzantısına bağlanacağınızı göstermektedir.

Son kısımda yer alan ve **Top Level domain** olarak geçen domain ismi kategorinize göre size otomatik olarak atanmaktadır.



Resim 3.6.1

NOT: Domain isimlerinin belirli veri sunucularında tutulması gerekmektedir. Örneğin bir şehirde alınan domain ismi oradaki sunucularda tutulmaktadır ve ayrıca bu sunucular, diğer domain ismi tutan sunuculara alınan ismin bilgisini vermektedirler. Böylece bir yerde

3.7 DNS (DIRECT NAME SERVER)

DNS (Direct Name Server) bilgilerinizi bilgisayarınızın terminalinden kolayca görebilmektesiniz. Bunun için terminalinize **nslookup** komutunu girmeniz ve daha sonra neye bakmak istediğinizi söylemeniz gerekmektedir. Örneğin şu an sunucuya (server) bakmak istediğimiz için nslookup yazdıktan sonra ayrıca server diye belirtmemiz gerekmektedir. Böylece bize bağlı olan isim sunucularımızın bilgisine ulaşabilmekteyiz. Komutun çalıştırılmış hali aşağıdaki gibidir:

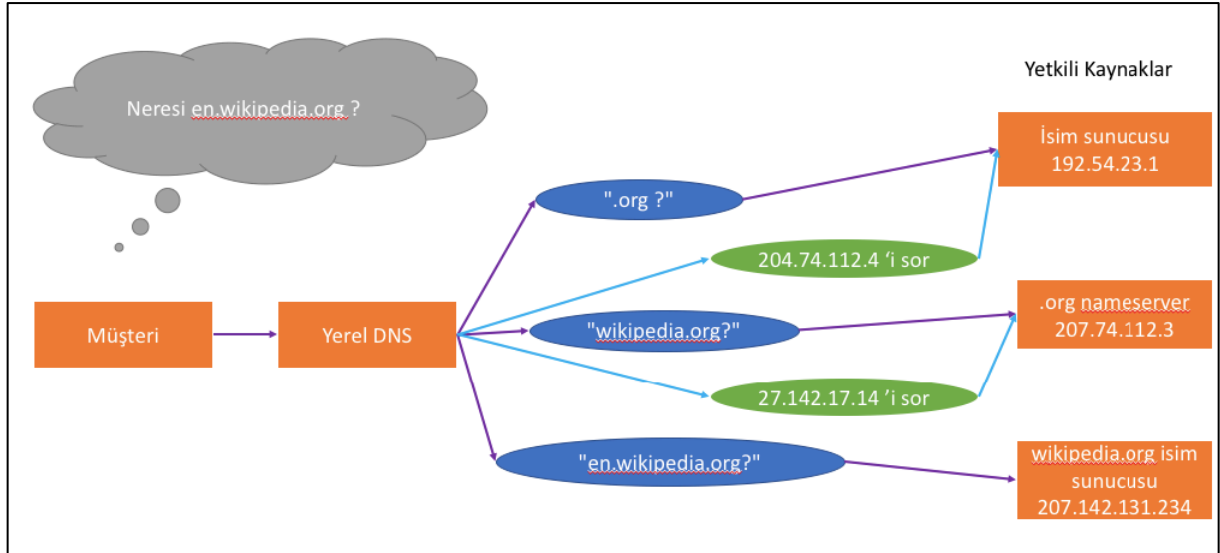
```
Last login: Sun Aug 5 13:02:16 on ttys000
[Bilgisayars-MacBook-Air:~ bilgisayaravramlari$ nslookup
[> server
Default server: 10.10.10.1
Address: 10.10.10.1#53
Default server: 208.67.220.220
Address: 208.67.220.220#53
Default server: 208.67.222.222
Address: 208.67.222.222#53
>
```

Resim 3.7.1

NOT: Eğer isterseniz daha sonra herhangi bir sitenin adresini girebilir ve onun bilgilerine erişebilirsiniz. Ya da bu komutları kullanarak sunucunuzu değiştirebilmeniz mümkündür.

3.7.1 DNS NASIL ÇALIŞIR?

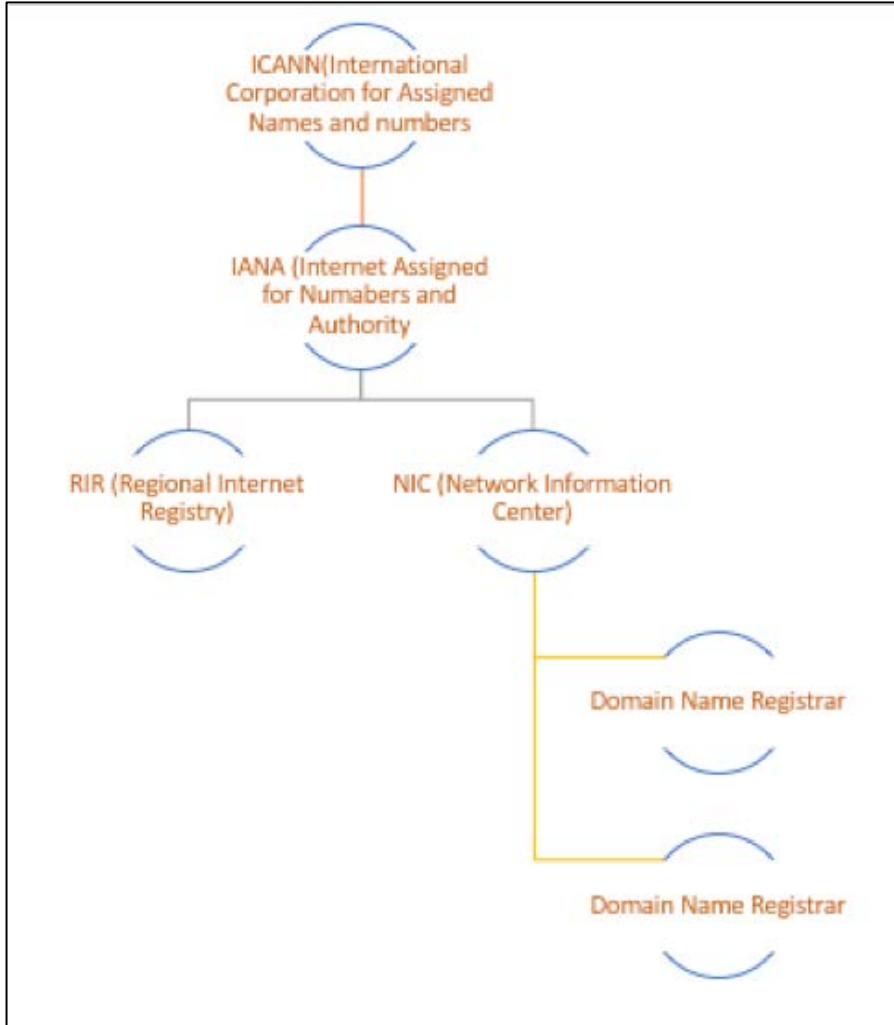
Bir web sitesine bağlanırken, bizim IP adresimizden çıkan veri, ilk olarak yerel sunucuya giderek bu sitenin IP adresini ister. Yerel sunucu onu domain isminin bulunduğu sunucuya yönlendirir. Daha sonra **top level domain** ve en son da **subdomain** sorgularır. Veri belirli sunuculara yönlendirilerek yolunu bulur. Resim 3.7.1.1 de verinin akış şeması gösterilmiştir.



Resim 3.7.1

3.8 İNTERNET POLİTİKALARI

İnternette en alt basamakta bulunan domain ismi sağlayıcıları belirli kuruluşlara bilgilerini iletmek zorundadırlar. Ve onların bağlı olduğu **NIC** adı verilen kuruluş da başka bir katmana bilgi vermektedir. Bu şekilde katmanlardan oluşan bir internet politikası mevcuttur ve her katmanda ilgili şirket ya da kuruluşlardan oluşmaktadır. Resim 3.8.1 de bu kuruluşların isimleri ve katman şeması sırası ile verilmiştir.



Resim 3.8.1

NOT: Bu tür kuruluşların amacı internet ile ilgili meydana gelen sorunlara çözüm bulmaktır. Aynı şekilde bu kuruluşlara benzeyen fakat şemada yer almayanlar arasında **İEEE**, **İETF**, **İESG**, **RFC** gibi kuruluşlarda aralarında yer almaktadırlar.

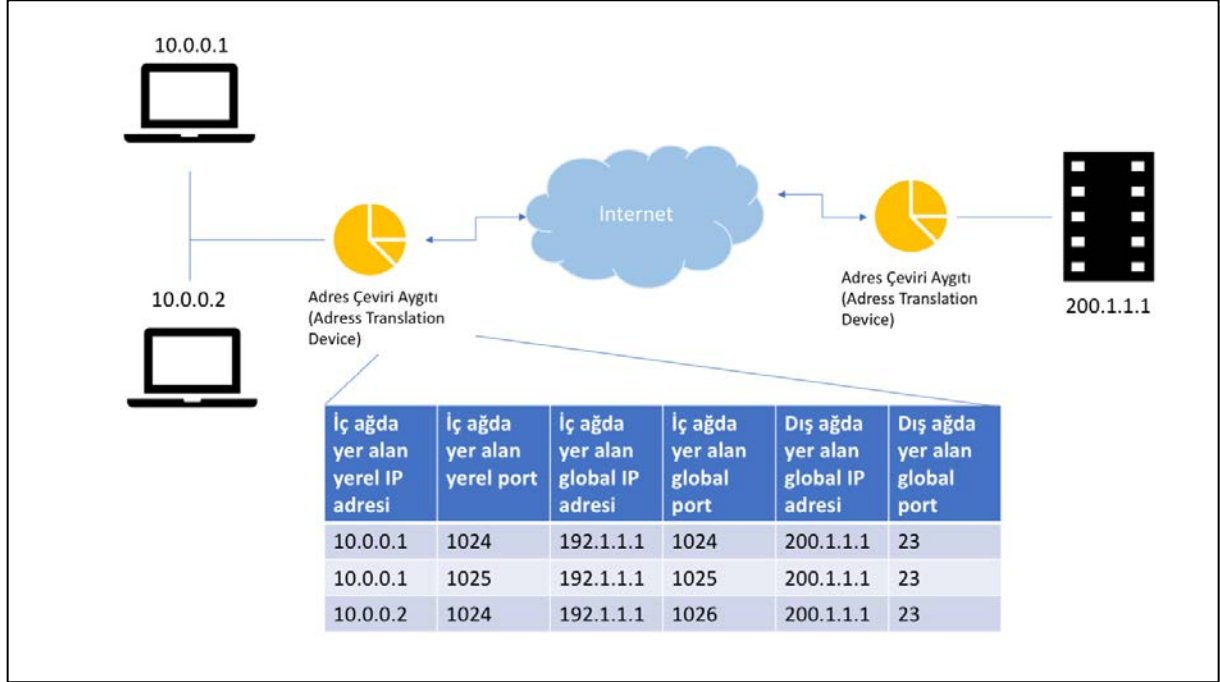
4.BÖLÜM: NAT

4.1 NAT HAKKINDA

4.2 NAT NASIL ÇALIŞIR?

4.1 NAT HAKKINDA

NAT olarak adlandırdığımız ağ adresi çevirisini, aynı zamanda **PAT (Port Address Translation)** olarak da görmemiz mümkündür. Bu yapının çalışma şeması Resim 4.1.1 de gösterilmiştir.



Resim 4.1.1

4.2 NAT NASIL ÇALIŞIR?

Yapıda ilk olarak bir yönlendirici bulunmaktadır ve bu yönlendiriciyi bir ADSL modeliniz olarak ya da bir firmanın, bir kurumun herhangi bir yerin çıkış kapısı olarak düşünmeniz mümkündür.

Aynı şema üzerine bir de internetin sol kısmında bulunan bir servis sağlayıcımız vardır. Servis sağlayıcısı herhangi bir sitenin bize bağlantısını sağlar. Yönlendiriciden diğer bilgisayarlara dağılan bağlantılar, port numaralarını tutarak IP numaralarını dışarıya dağıtmaktadır.

Resim 4.1.1 üzerinde yazılan IP adresleri her ne kadar sanal IP adresleri olsa da, onlara uygun olarak yazılan özel IP sağlayıcıları da bulunmaktadır. Dağıtmış olduğunuz IP adreslerini istediğiniz şekilde verebilmektesiniz. IP adreslerinin bilgilerini Resim 4.1.1 de tablo üzerinde de görüldüğü üzere yönlendiriciler tutmaktadır.

Örneğin tablodaki bilgilerden yola çıkarak, resimde bulunan A bilgisayarının herhangi bir siteye bağlanacağını düşünürsek, öncelikle IP bir name server tarafından bu sitenin IP adresine çevrilmesi gerekmektedir. Döngü yönlendiriciye geldiği zaman A bilgisayarını, belirttiği web sitesine bağlanmak istiyor ve beraberinde port bilgisini de iletiyor.

- **NOT:** Her IP adresi portlara yani sanal veri yollarına bölünmüştür. Bu sayede aynı anda, aynı IP adresinden (bilgisayar vb.) farklı programlarla veri alışverişi yapılabilmektedir. Örneğin aynı anda, aynı bilgisayardan 110 numaralı portu kullanarak Outlook Express'le e-maillerimizi indirirken, 80 numaralı portu kullanarak Internet Explorer ile web sitelerine bakabilmekteyiz. Port numaraları 0-65535 arasında değerler alabilmektedir.
- **NOT:** Bir web sitesinin sadece IP adresini söylemek o adrese bağlanmak için yeterli olmamaktadır. Aynı IP adresi üzerinde birden fazla hizmet sağlayıcısının portu bulunabilmektedir. Dolayısıyla bilgiler arasında kaçınıcı porta bağlanmak istediğimizin de belirtilmesi gerekmektedir.

Bağlanma işlemi gerçekleşirken istenilen cevapların iletilebilmesi için aynı zamanda web sitesinin de karşı tarafın port adresini bilmesi gerekiyor. Dolayısıyla A bilgisayarını cevabı hangi porttan alacak ise portun bilgisini web sitesine (servis sağlayıcısına) iletiyor.

İnternette alınacak olunan bilgiler cevap olarak geri iletilirken, gönderileceği port adresi PAT üzerinde başka bir port olarak daha tanımlanmaktadır. Çünkü PAT üzerinde birden çok cevap bulunabilir ve PAT onlara kendisi de ayrıca bir port tanımlayınca cevapları doğru bir şekilde teslim edebilmektedir.

- **NOT:** Örneğin bir bilgisayar NAT üzerinden tüm portları (65535 tane portu) kullanıyorsa, dolayısıyla başka bilgisayarlara vb. araçlara onun üzerinden bir yönlendirme hizmeti sağlamamaktadır. Dolayısıyla NAT daha çok küçük işler için iyi hizmet sunmaktadır.
- **NOT:** NAT, Yönlendirici için ayrı bir maliyet gerektirmektedir ancak eğer istersek iki tane Ethernet edinerek, birini yönlendirici olarak kullanabilmekteyiz.

5.BÖLÜM: AĞ PROTOKOLLERİ

5.1 AĞ STANDARTLARI

5.2 KATMANLAR VE PROTOKOLLER NASIL
İŞLER?

5.3 THREE WAY HANDSHAKE

5.4 DOS SALDIRILARI

5.5 DOSYA GÖNDERİMİ

5.1 AĞ STANDARTLARI

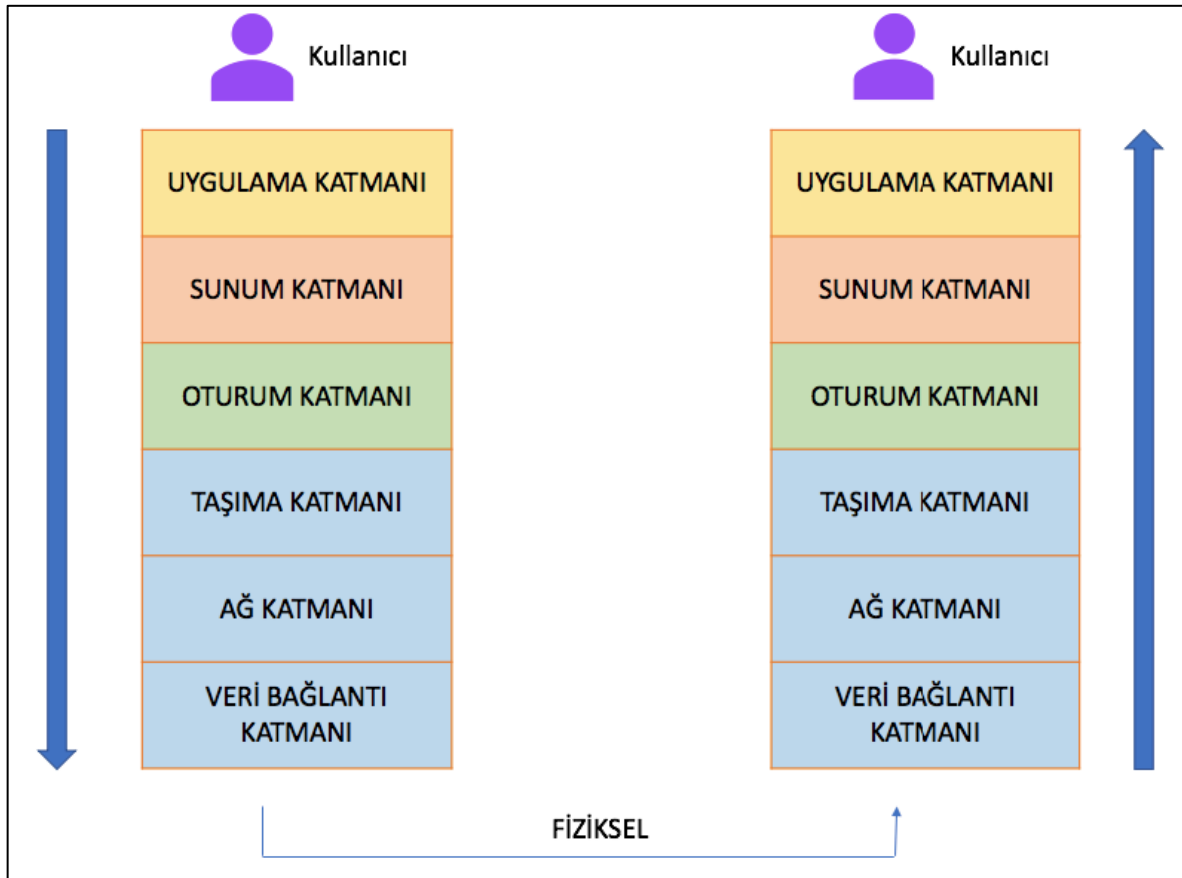
İnternet kullanılarak haberleşmeye başladıkları ilk yıllarda iki bilgisayarın birbirleriyle haberleşmeleri için aynı marka/model cihazları kullanmaları gerekiyordu. Bu yüzden üretilen bilgisayarların birbirleri ile sorunsuz çalışabilmesi için bazı standartlar geliştirilmiştir. Bu standartlardan en çok kullanılan iki tanesi **OSI** ve **TCP/IP**'dir. OSI, Açık Sistem Bağlantıları komitesi tarafından geliştirilmiş olup, TCP/IP ise Amerika Savunma Bakanlığı tarafından geliştirilmiştir.

OSI modelinde haberleşmenin sorunsuz çalıştırılabilmesi için 7 katmanlı bir ağ sistemi önerilmiştir.

TCP/IP'de ise bu haberleşmeyi sorunsuz çözebilmek için 4 katmanlı bir sistem oluşturulmuştur.

5.1.1 OSI (OPEN SYSTEM INTERCONNECT)

Çeşitli üreticilerin ürünlerinin bağlanabileceği bir ağ için Açık Sistem Bağlantıları komitesi tarafından geliştirilmiştir. OSI modeli herhangi bir bilgisayar ağı ya da donanıma göre değişiklik göstermemektedir.



Şekil 5.1.1.1

5.1.1.1 FİZİKSEL KATMAN

Fiziksel katman, verinin bitler halinde aktarıldığı katmandır. Bu katman 1 ve 0'lardan oluşan sayıların elektrik, ışık veya radyo sinyallerine nasıl dönüşeceğini ve aktarılacağını tanımlar. Bu katmanda herhangi bir protokol bulunmamaktadır.

5.1.1.2 VERİ BAĞLANTI KATMANI

Veri bağlantı katmanı, fiziksel katmana erişmek ve kullanmak ile ilgili kuralları belirlemektedir. Veri bağlantı katmanına gelen veriler fiziksel katmana aktarılırlar. Aktarım yöntemleri verileri kendi protokollerine uygun olarak işleyerek iletirler. Ayrıca veri bağlantı katmanında veriler iletilirken parçalara bölünürler. Bu katmanda bulunan protokollere örnek olarak **ATM**, **PPP** ve **HDLC** verilebilir.

5.1.1.3 AĞ KATMANI

Ağ katmanı, verinin farklı bir ağa gönderilmesi gerektiğinde gerekli bilgilerin sağlandığı ve verinin paketler halinde gönderildiği katmandır. Verinin en kısa yoldan hedefe ulaştırılmasını sağlamakla görevlidir. Bu katmanda bulunan protokoller örnek olarak **IP**, **ARP**, **ICMP**, **RARP** ve **BOOTP** verilebilir.

5.1.1.4 TAŞIMA KATMANI

Taşıma katmanı, üst katmanlardan gelen verinin nasıl taşınacağını belirleyen katmandır. Üst katmanlara taşıma hizmeti sağlamakla birlikte hizmetin kalitesini de artırmaktadır. Bu katmanda yer alan protokollere örnek olarak **TCP** ve **UDP** verilebilir. Bu protokoller aynı zamanda veride yer alan hataların kontrolünü de sağlamaktadırlar.

5.1.1.5 OTURUM KATMANI

Oturum katmanı, iki bilgisayar arasında yer alan uygulamalar arasındaki oturumun açılmasını, kullanılmasını ve bitirilmesi sağlayan katmandır. Bir bilgisayarın birden fazla bilgisayarla iletişim kurması gerektiğinde, doğru adresle iletişim kurmasını sağlamaktadır. Bu katmanda yer alan protokollere **SQL**, **NFS** ve **NetBios** verilebilir.

5.1.1.6 SUNUM KATMANI

Sunum katmanı, verinin sunum şeklini belirleyen katmandır. Bu katmanın en önemli görevi, alıcı bilgisayara verinin anlaşılabilir şekilde iletilmesini sağlamaktır. Böylece farklı türden bilgisayarların birbirleri arasında veri iletebilmesini sağlanabilmektedir. Sunum katmanı aldığı veriyi uygulama katmanına ileterek verinin formatının belirlenmesini sağlamaktadır. Bu katmandaki protokollere örnek olarak **JPG**, **MPEG**, **GIF** ve **ASCII** verilebilir.

5.1.1.7 UYGULAMA KATMANI

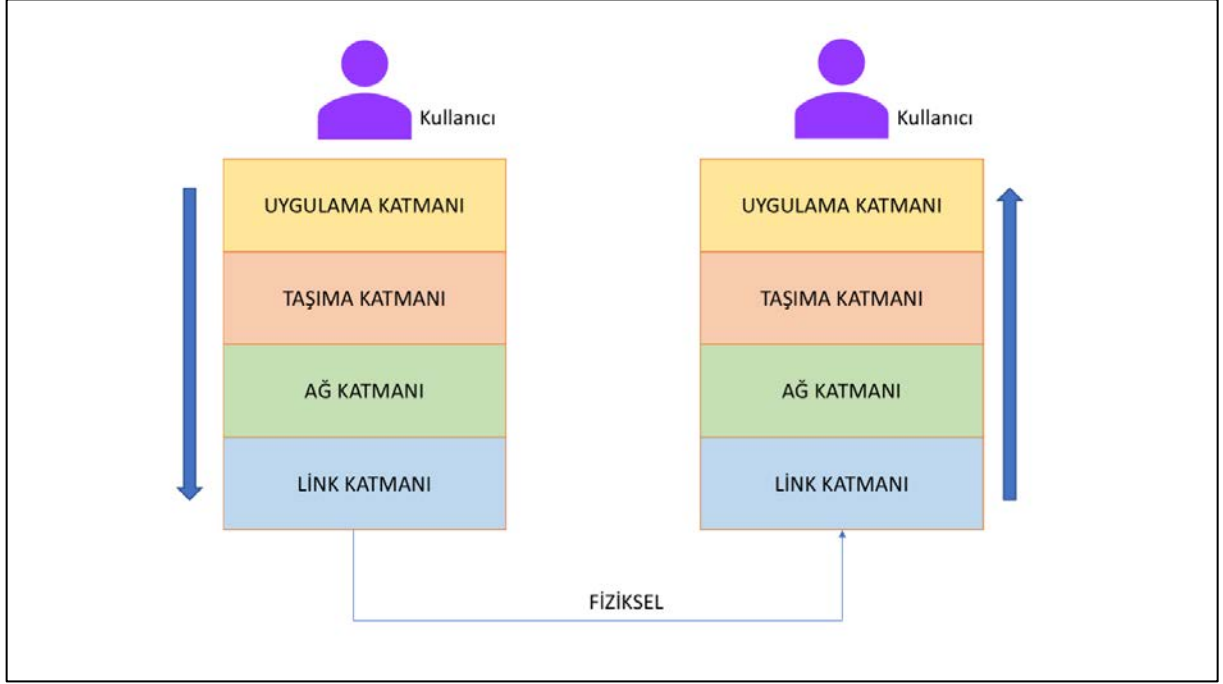
Uygulama katmanı, bilgisayar ile ağ arasında bir arabirim oluşturmaktadır ve programların ağı kullanabilmesini sağlamaktadır. OSI katmanları arasında sadece bu katman diğer katmanlara aktarım yapmaz.

Uygulama katmanı, katmanlar arasında kullanıcıya en yakın olanıdır. Bu katmanda yer alan protokollere örnek olarak **HTTP**, **TFTP**, **DNS**, **Telnet** verilebilir. Ayrıca tarayıcılarda bu katmanda çalışmaktadırlar.

E-Posta ve veri tabanı gibi uygulamaların gereksinimi bu katman aracılığı ile sağlanmaktadır.

5.1.2 TCP/IP

TCP/IP modelinde katmanların en üstünde kullanıcı bulunmaktadır ve kullanıcılar bilgileri oluşturarak, daha sonra da elektrik sinyaline dönüştürerek diğer katmanlara iletmektedirler.



Resim 5.1.2.1

Resim 5.1.2.1 deki şemada da gösterildiği üzere kullanıcıdan alınan bilgi, elektrik sinyaline dönüştürülerek ve bu şekilde diğer kullanıcıya aktarılarak bağlantı sağlanmaktadır.

Bağlantıları katman katman yapmanın amacı, üst katmanların alt katmanlardakiler ile bağlantısını kesmektir. Örneğin şemadan yola çıkarak, taşıma katmanı uygulama katmanından aldığı bilgiyi ağ katmanına iletir. Böylece aslında ikisi arasında bir aracı görevi görmüş olur. Fakat katmanların görevi bir tek bununla da sınırlı kalmaz. Örneğin taşıma katmanı, diğer noktada bulunan taşıma katmanı ile de konuşarak paketlerin iletilip iletilmediğini, eksik olup olmadıklarını da kontrol eder.

Her katman kendine eş olan diğer bir katmanla konuşarak, görevine uygun olan kontrolleri yaparlar. Diğer bir örnek olarak ağ katmanı 'paketlerin nasıl yolunu bulacağı' ile ilgili soruları sorarken, link katmanı Ethernet, fiziksel katman elektrik ile ilgili kontrollerini yapmaktadır.

5.1.2.1 LİNK KATMANI

Gidilecek olan yolu belirlemek link katmanı önemlidir. Ethernet bağlantısı gibi düşünülebilir. Bir ağa bağlandığımız zaman, her bilgisayarın kendi üzerinde paketler taşıması söz konusudur.

Link katmanı, iletişim ortamının özelliklerini, haberleşme hızını ve kodlama şemasını belirlemektedir. Bu katmanda çalışan protokollere örnek olarak **Ethernet, Wi-Fi, Token Ring** ve **ATM** verilebilir.

5.1.2.2 AĞ KATMANI

IP katmanı olarak da adlandırılan bu katman, verilerin gönderilmesini ve yönlendirilmesini sağlamaktadır.

IP (Internet Protocol) adresinin ağ için önemli olduğunu daha önceki bölümlerde belirtmiştik. Örneğin Dünya'nın bir ucundan diğer ucuna IP adresinin nasıl bir yol izleyeceği ağ katmanı tarafından kontrol edilmektedir.

NOT: Görselleştirilmiş olan katmanlar TCP protokolüne ait olan katmanlardır. Ayrıca bir de OSI katmanları bulunmaktadır.

5.1.2.3 UYGULAMA KATMANI

Bağlantı kurabilmek için bir uygulama çalıştırmamız gerekmektedir. Örneğin bir web sunucusuna bağlanacaksanız, kendi iletişim aracınızdaki web gezgini ne ise, karşı tarafta onunla bağlantıya geçilmesini uygulama katmanı sağlamaktadır. Eğer farklı bir uygulamaya paket gönderilirse cevap alınamaz çünkü iki uygulamanın karşılıklı bağlantıya geçtiği protokol ile diğerleri aynı değildir. Örneğin **Http (Hyper Text Transfer Protocol)** bir web sitesi protokolüdür ve dolayısıyla bağlantıya geçeceği uygulamalar ile Http protokolünün kontrolü gerçekleştirilir.

5.1.2.4 TAŞIMA KATMANI (TCP – TRANSPORT CONTROL PROTOCOL)

Kitabın bu kısmında özellikle **TCP (Transfer Control Protocol)** protokolünden bahsedilecektir fakat yaygın olarak bilindiği üzere **IP (Internet Protocol)** ‘ünün üstünde ayrıca **UDP (User Datagram Protocol)** gibi bir protokol de çalışabilmektedir. Fakat TCP’nin UDP’den farkı güvenilir bir protokol olmasıdır. Göndermiş olduğunuz paketlerin hepsinin karşı tarafa ulaştığını garanti etmektedir. Fakat UDP gönderdiği paketlerin karşı tarafa gönderildiğini garanti etmemektedir.

TCP genel olarak yüksek alana sahip olan verileri 1500 B paketlere bölme işlemini gerçekleştirmektedir. Daha sonra her paketi bir sıraya koymaktadır ve bu paketlerin gideceği portları ayarlamaktadır. Son olarak tıkanıklık (congestion) kontrolü yapılmaktadır. Örneğin paket gönderdiğiniz yerde paket yoğunluğu var ise ya da göndermeniz gereken daha fazla paket var ise TCP bu tıkanıklığı algılayarak paketlerinizi yavaşlatma imkanına sahiptir. Böylelikle karşı tarafa gönderilecek olan paketin kaybolmamasını sağlamaktadır ve paketler karşı tarafa doğru bir şekilde ulaştırılmaktadır.

5.2 KATMANLAR VE PROTOKOLLER NASIL İŞLER?

Bir web sayfasının ekranınıza gelene kadar izlediği yol aşağıda verilmiştir.

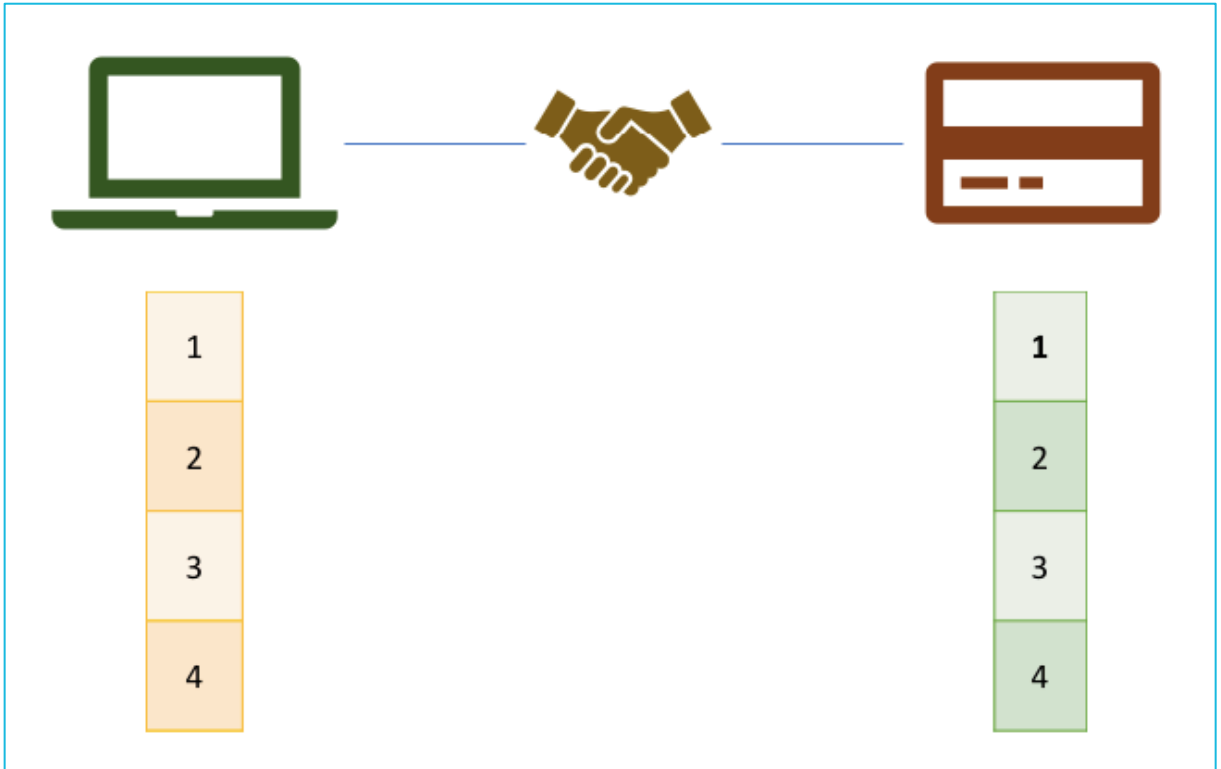
- Web sayfasının bulunduğu sunucuda, uygulamalar sayfanın HTML formatında bir çıktısını oluşturur. HTML formatında çıktısı oluşturulan veriye, HTTP protokolü ile gönderilmesi komutu verilir. Uygulama katmanında gerçekleşen bu adımlardan sonra veri, taşıma katmanına aktarılmaktadır.
- Taşıma katmanında veriye bu katman ile ilgili veriler (port bilgisi ve veri boyutu gibi) eklenmektedir.
- Ağ katmanına ulaşan veriye, verinin gönderileceği bilgisayarın (sunucunun) ve sizin bilgisayarınızın IP adresleri eklenmektedir.
- Fiziksel katmanda fiziksel adresler ve verinin son halinin boyutu eklenir.
- Veri paketi sunucudan çıkar ve sunucu ile bilgisayarınız arasındaki yolu takip ederek bilgisayarınıza ulaşır.
- Veri bilgisayarınıza ulaştığı zaman, bu sefer sıralamaya tersten başlayarak (Fiziksel – Ağ – Taşıma – Uygulama) katmanlardaki protokoller işletilir. En sonunda veri paketi web tarayıcınıza gönderilir ve işlem tamamlanır.

NOT: Her katmanda ayrı donanımlar kullanılır. Örneğin fiziksel katmanda **Switch**, ağ katmanında **Router**, taşıma katmanında ise **NAT** gibi donanımlar kullanılmaktadır.

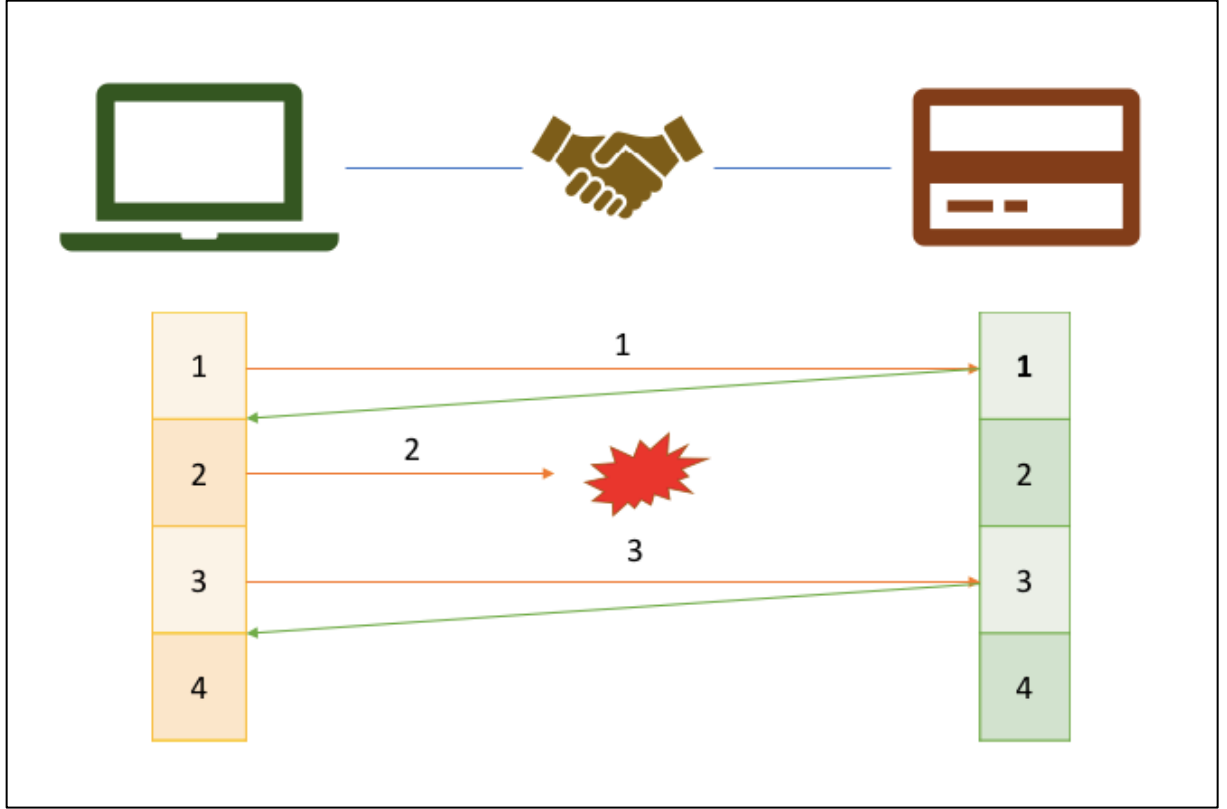
NOT: Her katmanda ayrı donanımlar kullanılır. Örneğin fiziksel katmanda **Switch**, ağ katmanında **Router**, taşıma katmanında ise **NAT** gibi donanımlar kullanılmaktadır.

5.3 THREE WAY HANDSHAKE

Handshake (Türkçeye el sıkışmak olarak çevrilebilir), TCP konusunda çok yaygın olarak verilen bir örnektir. Örneğin bir siteye bağlanmak istiyorsunuz ve dolayısıyla o siteye paketleri gönderiyorsunuz. Herhangi bir problemten dolayı gönderemeyebilirsiniz ya da gönderdiğiniz fakat paket ulaşılmamış olabilir. Ya da 3. bir ihtimal olarak gönderdiğiniz fakat paket ulaşıldı şeklinde bir geri dönüş size ulaşmadı. Bu 3 kontrol için ayrı ayrı protokol gerektirmektedir ve buna da **Three Way Handshake** denilmektedir. Buna göre hem ağ kurulum aşamasında hem de ağı bitirirken Three Way Handshaking mekanizması çalıştırılmaktadır.



Resim 5.3.1



Resim 5.3.2

Three Way Handshake çalışma mekanizması, Resim 5.3.2'den yola çıkılarak açıklanacaktır. Görüldüğü üzere ilk paketin gönderilmesinde bir sorun yokken, ikinci paketin geldiğine dair geri bildirim bir problemden dolayı gönderilememiş. Dolayısıyla biz üçüncü paketi gönderdikten sonra, paketleri alan taraf durumu fark ederek iki paket birden gönderiyor ve birinci paketin geldiğine dair geri bildirimini tekrar istiyor. Böylelikle Three Way Handshake mekanizması sayesinde TCP paketlerin ulaştığına dair garanti verebilmektedir.

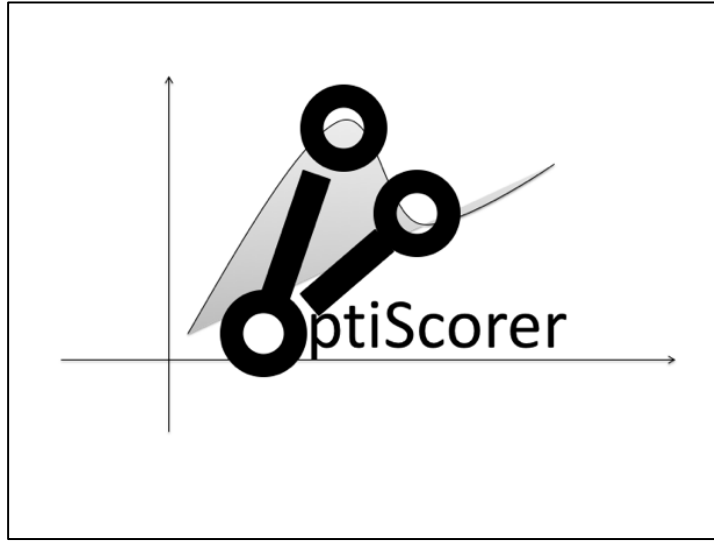
5.4 DOS (DENIAL OF SERVICE ATTACKS) SALDIRILARI

Denial of Service (DOS) hizmet aksatma amaçlı bir saldırı çeşididir. Bir sisteme yapılan düzenli saldırılar sonucunda sistem çalışamaz ve hizmet veremez hale gelir. DOS genellikle hedef makine veya kaynağın, gereksiz talepler ile aşırı yüklenmesi ve bazı ya da bütün meşru taleplere doluluktan kaynaklı engel olunması şeklinde gerçekleştirilir. Ayrıca DOS saldırılarıyla hedef sisteme ait kaynakların tüketilmesi de amaçlanır ve bu saldırı önemli sunucularının servis vermeyi durdurması gibi büyük sorunlara yol açabilir.

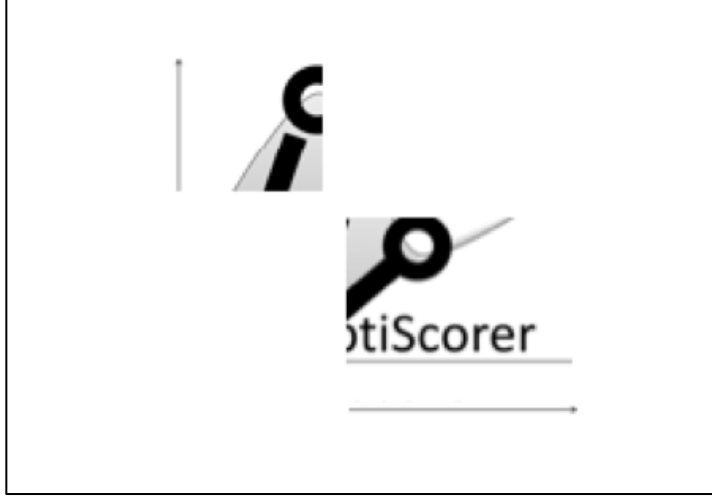
5.5 DOSYA GÖNDERİMİ

TCP üzerinden dosya gönderiminde öncelikle bu dosya boyutunu 1500 B'den daha küçük parçalara bölünmektedir. Daha sonra bir IP header (başlık) eklenerek gönderilmektedir. Her gönderme işlemi bir **çerçeve (frame)** üzerinden sağlanmaktadır. Bu çerçeve üzerinde ekleme veya çıkarma işlemi yapılabilir fakat sizin veriniz ve onun üzerine eklenen TCP bilgisi sabit olarak kalır. Ondan sonra bu bilgi alınan taraf tarafından, uygulama bölümünde açılabilir.

Örneğin Resim 5.5.1, başka bir yere taşınırken Resim 5.5.2'deki gibi gösterilmiştir.



Resim 5.5.1



Resim 5.5.2

Örneğin bu resmi taşıırken bir eksik oldu ya da hatalı bir veri geldi, bu sorunları bulup düzeltmek TCP'nin görevidir.

6.BÖLÜM: WORLD WIDE WEB

6.1 WWW TARİHİ

6.2 BROWSER REKABETLERİ

6.3 WEB STANDARTLARI

6.4 URL

6.5 İSTEMCİ – SUNUCU İLİŞKİLERİ

6.6 TELNET

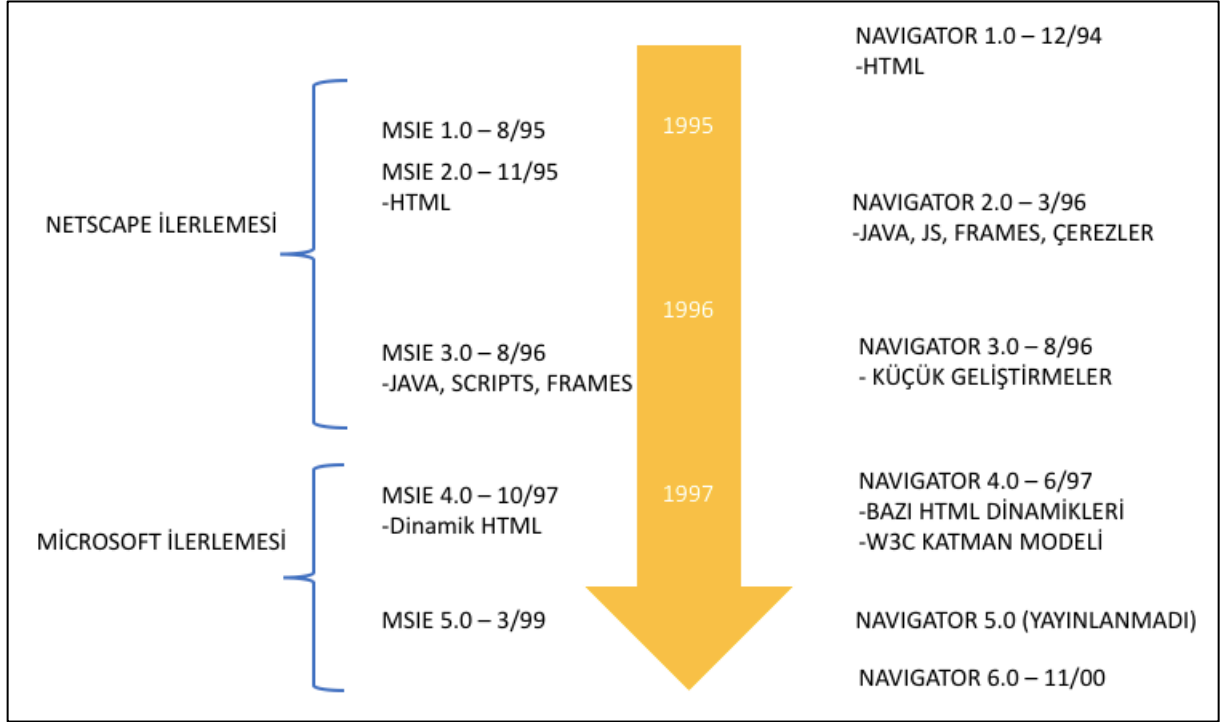
Daha önceki bölümlerde internetin temel altyapısının nasıl çalıştığından bahsetmiştik. Kitabımızın bu bölümünde uygulama katmanına giriş yapılacaktır. Bu yapı Firefox, Opera, Chrome ya da Safari gibi tarayıcıların çalıştığı bir yapıdır.

6.1 WWW TARİHİ

- 1945 yılında Vener, **MEMEX** teorisini ortaya atmıştır.
- 1967 yılında ise Ted Nelson **Hypertext** adında metin üstü bir bağlantı anlamını taşıyan kavramını öne çıkarmıştır.
- Daha sonra **Xanadu** adında bir liste yapılmış ve bu listede bir metnin nasıl tutulacağına dair maddelere yer verilmiştir. Xanadu aslında üretilen bir proje değildir yani gerçekte sadece bir bulut projedir. Ayrıca daha sonralarda **Hypercard** olarak devam etmiştir. Fakat bu projenin başarısız olduğu bilinmektedir çünkü kuralları çok sıkıdır ve donanımının kurulmasında başarılı olunamamıştır.
- 1990: Tim Berne-Lee ve Robert Calilliu ağı kullanmışlardır (CERNE'de bulunan Hypertext dokümanları için).
- 1991: İlk Amerikan sunucu, **SLAC** tarafından yapılmıştır.
- 1992: Super bilgisayar kavramı ve ardından **Mozaik browser** kavramları ortaya çıkmaktadır. Mozaik browser geliştiren kişiler Mac ve Windows'a **Netscape** adı altında bir browser vermişlerdir ve bu olay bazı sorunlara yol açmıştır.

6.2 BROWSER REKABETLERİ

Resim 6.2.1 üzerinde Netscape ve Microsoft arasında yaşanan browser rekabeti ve onlara bağlı olarak gerçekleşen gelişmeler şematize edilmiştir.



Resim 6.2.1

İnternet tarayıcılarında reklam yapan firmalardan ilki **Mosaic Communications Corp.** firmasıdır. Daha sonra ismini **Netscape Communications Corp.** olarak değiştirilmiştir.

Firmanın tarayıcısının ismi Mosaic Netscape 0.9 olarak çıkmıştır. Tarayıcı büyük ilgi görmüş olup, o zamanlarda Pazar payı %90'ları bulmuştur (Günümüzde %1 pazar payına sahiptir.).

Netscape'in yayınlanışından 6 ay gibi bir müddet sonra web trafiği, internetin en önemli trafiği haline gelmiştir.

Web trafiğinde yaşanan bu yoğun ilgiden dolayı Microsoft pazara dahil olmaya karar verdi ve **İnternet Explorer**'ın lisansını satın alarak rekabeti başlattı.

NOT: Herkes tarafından öyle biliniyor olsa da İnternet Explorer aslında tam olarak bir Microsoft ürünü değildir. Microsoft, İnternet Explorer'ın lisansını **Spyglass** adında bir firmadan satın almıştır.

Daha sonraki 14 yıl boyunca, Microsoft, İnternet Explorer lisansını elinde tutmuş olup, hala gün geçtikçe geliştirmektedir. Bu zaman zarfında Pazar payını %49'a yükselterek, Netscape tarayıcısının Pazar payının düşmesine sebep olmuştur.

Opera isimli Norveç asıllı bir firma, mobil aygıtlar için bir **İnternet Opera** yazılımını ve ayrıca Windows platformu için de bir yazılım geliştirmiştir.

Opera'nın ilk zamanlarda sadece 1.1'lik bir pazar payı olsa bile, şimdilerde Microsoft, Mac ve Linux platformlarından 40 milyon kullanıcısı bulunmaktadır.

Eski başarısını tekrar elde etmek isteyen Netscape, 1998 yılında **Mozilla** projesini hayata geçirdi ve başarılı oldu. Geliştirilmesine katkıda bulunduğu açık kaynak kodlu internet tarayıcısı **Firefox**, İnternet Explorer için büyük bir risk teşkil etmektedir.

1998 Yılıının sonlarına doğru Netscape, AOL tarafından 4.2 milyar dolara satın alındı fakat AOL, Netscape'in gelişimine uzun bir süre katkıda bulunamadığı için şimdilerde önemini yitirmiştir.

6.3 WEB STANDARTLARI

Hemen her sektörün kendine ait standartları olduğu gibi, web dünyasının da standartları vardır. Web standartları olarak adlandırdığımız kurallar bütünü internet üzerindeki bilgilerin, nasıl doğru bir şekilde kullanıcıya aktarılabilceğini, web sayfasının nasıl kodlanması gerektiğini ve tarayıcılarda nasıl görünmesi gerektiğini açıklamaktadır.

1994 yılında web teknolojilerinin standardize edilmesinde **World Wide Web Consortium (W3C)** çok etkili olmuştur. Bu yıllarda kurulan W3C, kendi Html ve Css adı altında kendi standartlarını belirlemiştir.

Ayrıca **ACID testi** adı verilen bir test geliştirmişlerdir ve bu test o zaman şartlarında teknolojinizin uygun olup olmadığını kontrol etmektedir. Bu testte o zamanların gelişmiş teknolojisi ile geliştirilen bir Html web sayfası bulunmaktadır. Eğer tarayıcınız bu sayfayı görüntüleyebilirse teknolojinizin o günkü şartlara uygun olduğu anlamına gelmektedir.

Html ve Css konusundan ileride yer alan bölümlerde ayrıntılı olarak bahsedilecek olup, kısaca açıklamak gerekirse;

6.3.1 HTML

Açılımı **Hyper Text Markup Language (Hareketli Metin İşaret Dili)** olup, bir web sayfası hazırlanırken temel kodlama Html ile yapılmaktadır.

W3C tarafından daha sonra geliştirilerek **XHTML** adını almıştır. Bu geliştirme arama motorlarının web sayfalarını daha iyi bir şekilde görüntüleyebilmesini sağlamak için yapılmıştır.

6.3.2 CSS

Açılımı **Cascading Style Sheets (Basamaklı Biçim Stil Şablonları)**'dir. Bir web sayfasında pozisyonun, başlığın, menünün, paragrafın, metnin ve tablonun nasıl görüneceğini belirleyen kuralları içermektedir.

Renklendirme, genişlik, yükseklik, metin fontu veya büyüklük gibi özellikler, web sayfalarının indekslenmesini sağlayan arama motorları için önemli bir gerekliliktir.

6.4 URL (UNIFORM RESOURCE LOCATORS)

URL açılımı **Uniform Resources Loader (Tekdüzen Kaynak Bulucu)** olan, bir web sitesine erişebilmemiz için kullanılan özel bir ID'dir. Bilgisayarların sunucular ile iletişim kurdukları ve sayıların kullanıcıların okuyacağı şekilde metine çevrilmesini sağlamaktadır.

Farklı ülkelerden aynı siteye bağlanan insanların, aynı adrese bağlanabilmelerini sağlayabilmeleri için geliştirilmiştir. Örneğin www.bilgisayarkavramlari.com sitesine Türkiye'den bağlanmak isteyen biri ile Çin'den bağlanmak isteyen birinin, farklı sitelere bağlanmasını engelleyebilmek için.

Aynı zamanda **HTTP**, **HTTPS** ve **FTP** gibi protokollerin nasıl alınacağını da belirtmektedir.

6.4.1 URL NASIL ÇALIŞIR?

URL çalışma prensibinde, ilk kısmına nasıl (**http**, **https**, **ftp** vb.) çalıştığı, daha sonra nereye bağlanacağı ve en son olarak da neye bağlanacağı (**.com**, **.edu** vb.) belirtilmektedir. Yani genel yapısı itibari ile bir protokol, bir domain adı ve bir yoldan oluşmaktadır.

nasıl:// nerede/neye

NOT: Domain adı bir web sitesinin bulunduğu konumun insanlar tarafından okunabilen kısmıdır.

Eğer URL’yi daha da özelleştirmek gerekirse, ilk kısma hangi protokolün uyulacağı, daha sonra host adı ve en son kısma ise dosya yolu belirtilmelidir.

protocoll:// host/dosyayolu

NOT: URL’ni host kısmına domain ismini ya da IP adresini de yazmanız mümkündür. Verdiğiniz IP adresi, bağlanmak istediğiniz web sitesine ait ise bağlanacaktır.

NOT: URL üzerinde # işareti koyarak sayfa üzerinde özel bir kısma gitmek mümkündür:
example.com/file.html#parttwo

NOT: Dinamik web siteleri ile birlikte URL üzerine gelen soru işareti (?) sembolü, bize bir parametre geçmesini sağlamaktadır.

http://www.mapguest.com /?q=bilgisayar+kavramları

6.5 İSTEMCİ – SUNUCU İLİŞKİLERİ

İstemci, web adreslerine bağlanma isteğinde bulunan ve **sunucu** ise bu hizmeti sunan sağlayıcıdır. Bundan yola çıkarak aralarındaki ilişki, aşağıdaki gibi açıklanabilir:

1. İlk olarak istemci bağlanmak istediği web sitesine tıklar
2. Web sitesinin IP numarası DNS üzerinden sorgulanır
3. DNS üzerinden edinilen IP adresine göre, TCP üzerinden hangi porta bağlanılacağı belirlenir
4. Bu yol ile istemci web sitesine bağlanıyor ve sunucu bu siteye bağlandığını onaylıyor
5. Daha sonra bir GET işlemi başlanılıyor ve istemci web sitesinin getirilmesini istiyor
6. Sunucu kendi diski üzerinden bu web sitesini arıyor
7. Bulduktan sonra bu web sitesini paketler halinde istemciye geri göndermektedir
8. Dosya transferi tamamlanıyor
9. TCP bağlantısı kapanıyor
10. İstemci web sitesini görüntü haline getirebiliyor

6.6 TELNET

Telnet, bir TCP/IP protokolüdür. İnternet ağına sahip bir bilgisayara, başka bir bilgisayarın bağlanması için geliştirilmiştir. Hem bu görevi yapan programlara, hem de protokole telnet adı verilmiştir.

Telnet, bağlantı için kullanılan terminallerden birisidir. Amacı size bir sunucu üzerinde konsol açarak, sanki oradaymışsınız gibi davranmaktır.

İstemci ve sunucu arasında yer alan bir protokol olan telnet, güvenli protokoller sınıfında yer almamaktadır çünkü bilgileri şifrelenmemiş olarak (**Düz Metin – Plain text**) aktarmaktadır.

Geliştirildiği zamandan bu yana farklı yeniliklere sahip olan telnet, teknolojinin gelişmesi ile birlikte bilgisayarlara işlemciyle gelmektedir. Böylece tekrar yüklenmesine gerek kalmamaktadır.

1969 Yılında geliştirilmeye başlanan telnet protokolü, o dönemde internet kullanıcılarının az olması ve alternatifinin olmaması nedeniyle devlet tarafından da kullanılmaktaydı. Daha sonraları gizli olması gereken bilgilerin şifrelenmemesi bir tehlike arz ettiği için kullanılmamaya başlanmıştır.

Günümüzde yaygın olarak kullanıldığı alan, **BBS (Bulletin Board Systems)** gibi sistemlere internet üzerinden erişilmek içindir. Aynı zamanda programlara verilen komutları test etmek ve hataları tespit etmekte kullanılmaktadır.

6.6.1 TELNET KOMUTU NASIL ÇALIŞIR?

1. Öncelikle kullanmış olduğunuz bilgisayarınızda terminal ekranını açmanız gerekmektedir.
2. ‘Komut’ satırına telnet ve bağlanılmak istenen makinenin IP adresi yazılarak ‘Enter’ tuşlanır.
3. Adres doğruysa komut satırı ekrana gelecektir.
4. Daha sonra bilgisayara bağlanmak için kullanıcı adı ve şifrenin girilmesi yeterlidir.

Örneğin bunu kendi terminalimizde örnekleyelim. İlk olarak **Telnet** komutu ile bir web sitesine bağlanıyoruz ve bu kısımda kaçınıcı porttan bağlanmak istediğimizi de belirtiyoruz. Daha sonra **GET** komutu ile birlikte home.html http/1.0 diyerek bilgilerine ulaşmış oluyoruz.

GET komutundan sonra belirttiğimiz bağlantı, http 1.0 protokolü ile istenen sayfaya ulaşılmasını sağlamaktadır. Anlatımın terminalde uygulanmış hali Resim 6.6.1'deki gibidir.

```
mb-sadi:~ sadievrenseker$ telnet amazon.com 80
Trying 54.239.17.6...
Connected to amazon.com.
Escape character is '^]'.
GET /home.html HTTP/1.0

HTTP/1.1 301 Moved Permanently
Date: Wed, 21 Sep 2016 15:24:33 GMT
Server: Server
Location: https:///home.html
Content-Length: 226
Keep-Alive: timeout=2, max=9
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https:///home.html">here</a>.</p>
</body></html>
Connection closed by foreign host.
mb-sadi:~ sadievrenseker$
```

Resim 6.6.1.1

7.BÖLÜM: HTML

- 7.1 HTML NASIL ÇALIŞIR?
- 7.2 UYGULAMA KURULUMU (MAMP)
- 7.3 HTML TARİHİ
- 7.4 HTML NEDİR?
- 7.5 GENEL BAKIŞ
- 7.6 ETİKETLER
- 7.7 HYPERLINK
- 7.8 KAÇIŞ KODLARI
- 7.9 HTML – LİSTELER
- 7.10 HTML – TABLOLAR
- 7.11 RESİM FORMATLARI

7.1 HTML NASIL ÇALIŞIR?

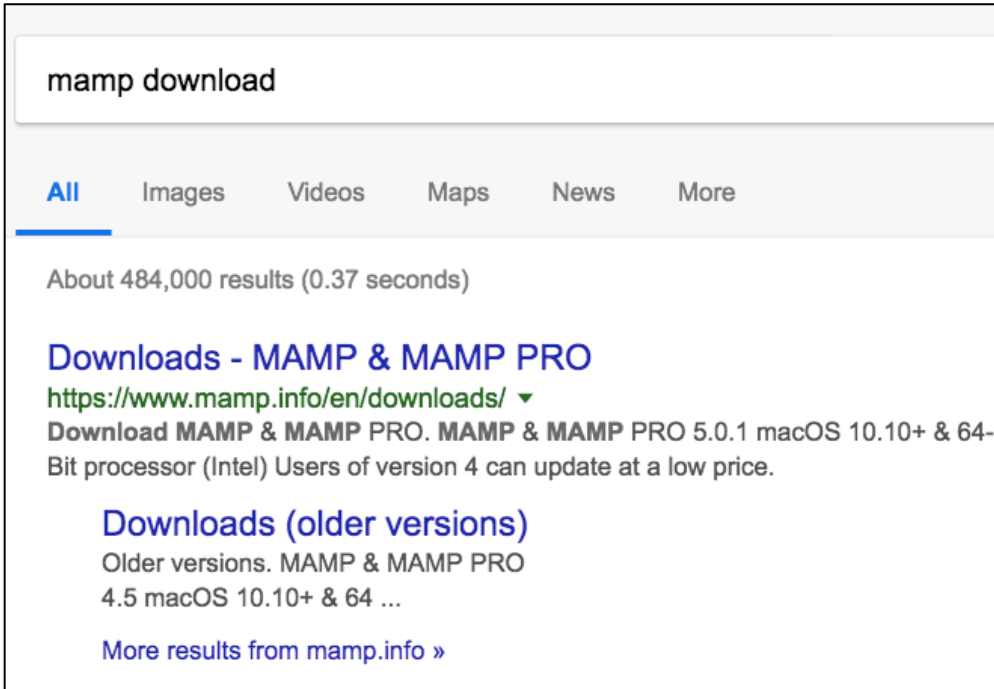
İnternet nasıl çalışır sorusunu anlayabilmek için, web sitelerinin nasıl çalıştığını da anlamak gerekmektedir. Bu yüzden web sitelerinin arka planında çalışan Html kodları nasıl çalışır? sorusunu bu bölümde açıklayacağız.

NOT: Html nasıl çalışır sorusunu açıklarken uygulama yapabilmek ve yapılan çalışmalarını gösterebilmek için Mac işletim sistemi için **MAMP** adı verilen uygulamayı kullanacağız. Bu uygulamanın Linux işletim sistemindeki ismi **XAMP**, Windows işletim sisteminde ise **WAMP** olarak geçmektedir. MAMP uygulaması bilgisayarımıza yerel bir web sunucusunun kurulmasını sağlamaktadır.

7.2 UYGULAMA KURULUMU (MAMP)

MAMP uygulamasının kurulum aşamalarını adımlar halinde belirtilmiştir:

1. İnternet tarayıcınızdan “Mamp download” yazarsanız, ilk sırada çıkan <https://www.mamp.info/en/downloads/> adresine girin.



Resim 7.2.1

2. Sayfa üzerinde size uygun işletim sisteminize göre **indir (download)** butonuna basarak, bilgisayarınıza indirme işlemini başlatabilirsiniz.



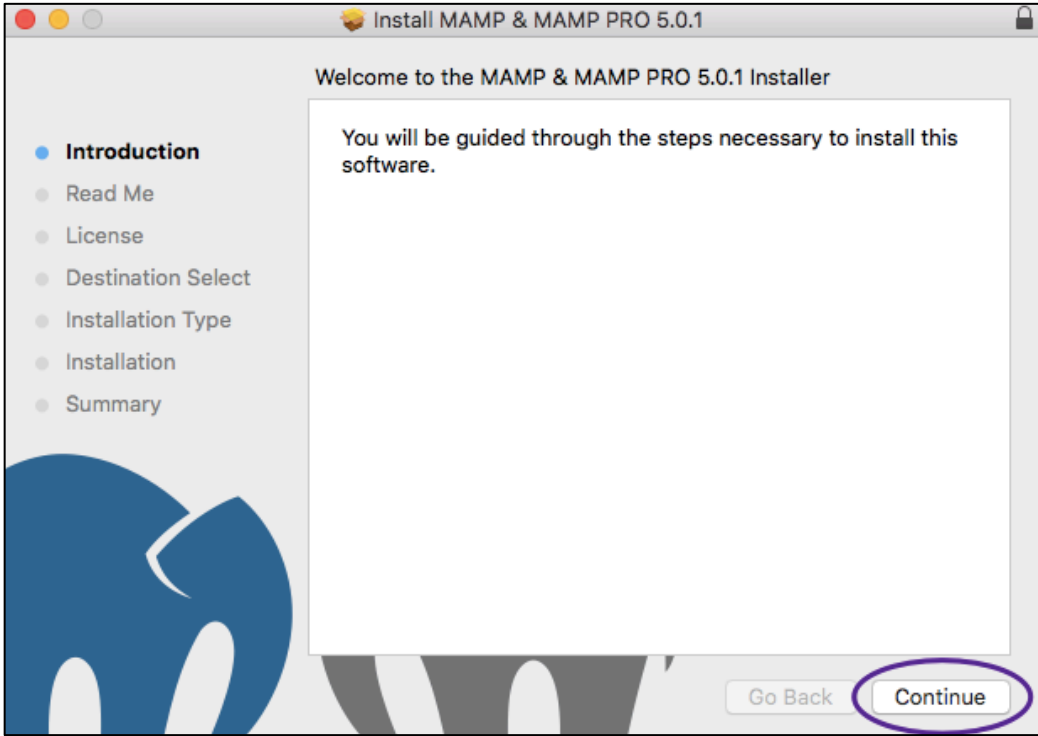
Resim 7.2.2

3. Bilgisayarınıza indirilen yükleme bağlantısına çift tıklayınız.

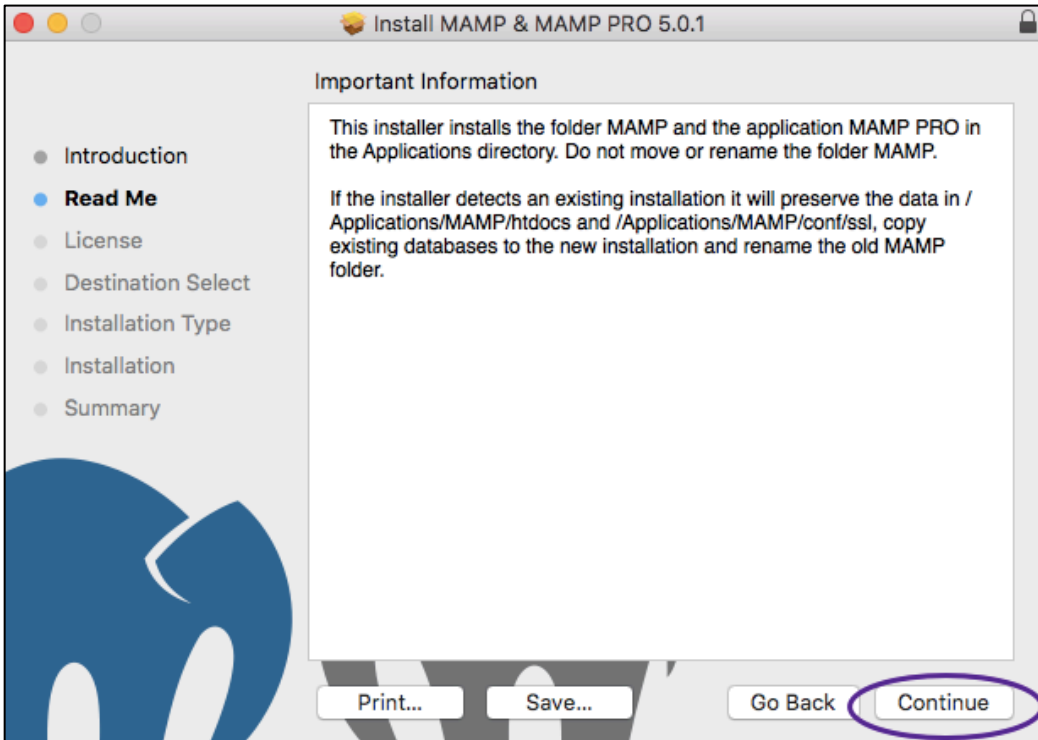


Resim 7.2.3

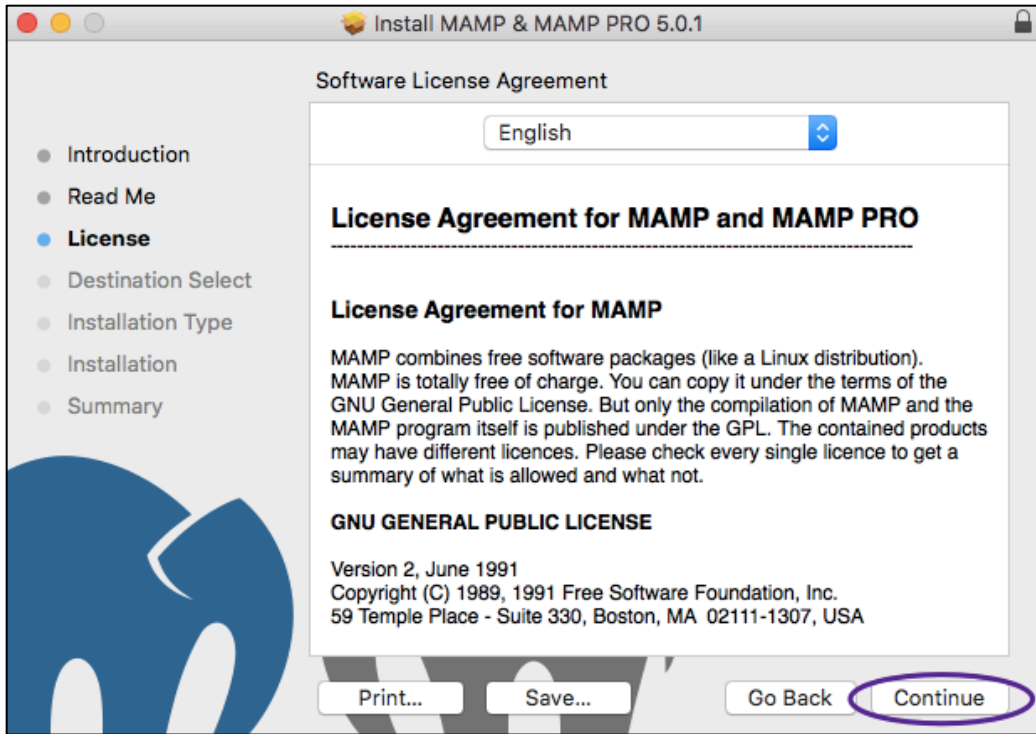
4. Ekranda açılan pencerede **devam (continue)** butonlarına tıklayarak devam ediniz.



Resim 7.2.4

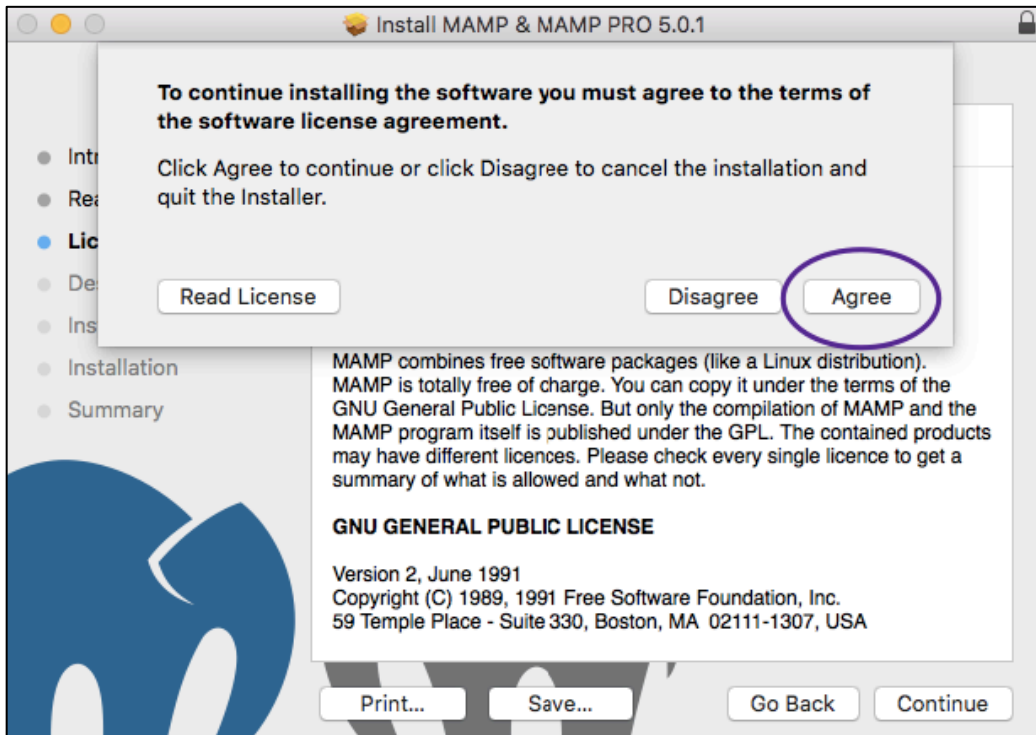


Resim 7.2.5



Resim 7.2.6

5. Açılan pencerede MAMP uygulamasına ait lisans anlaşmasını kontrol ve kabul ederek devam ediniz.

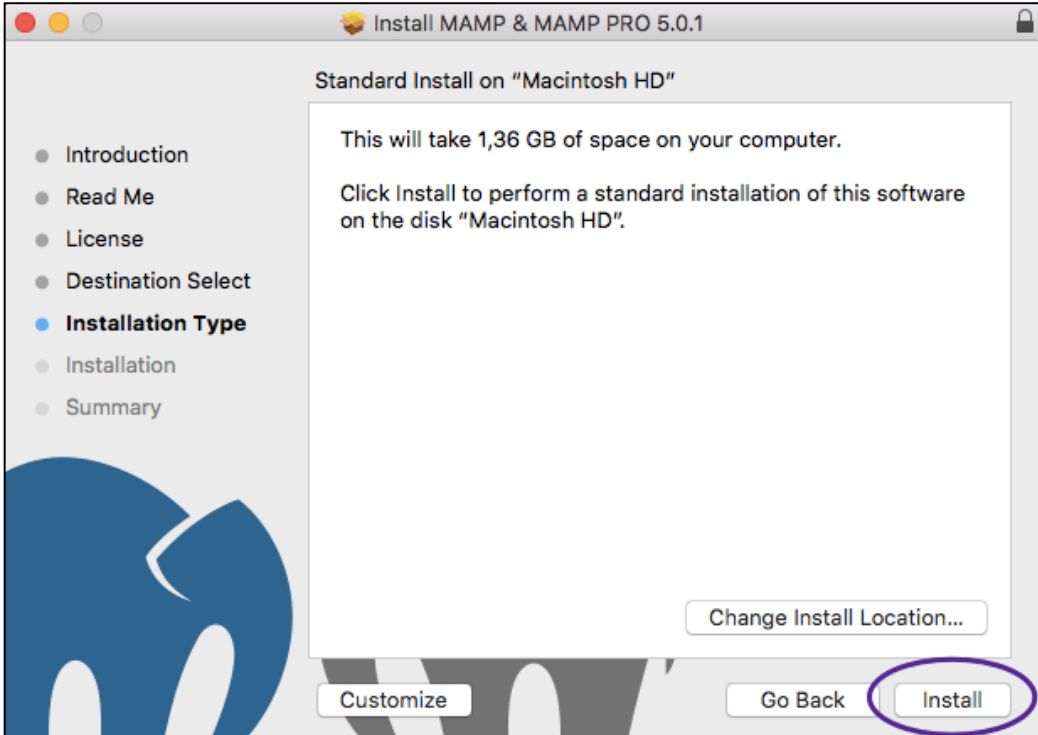


Resim 7.2.7



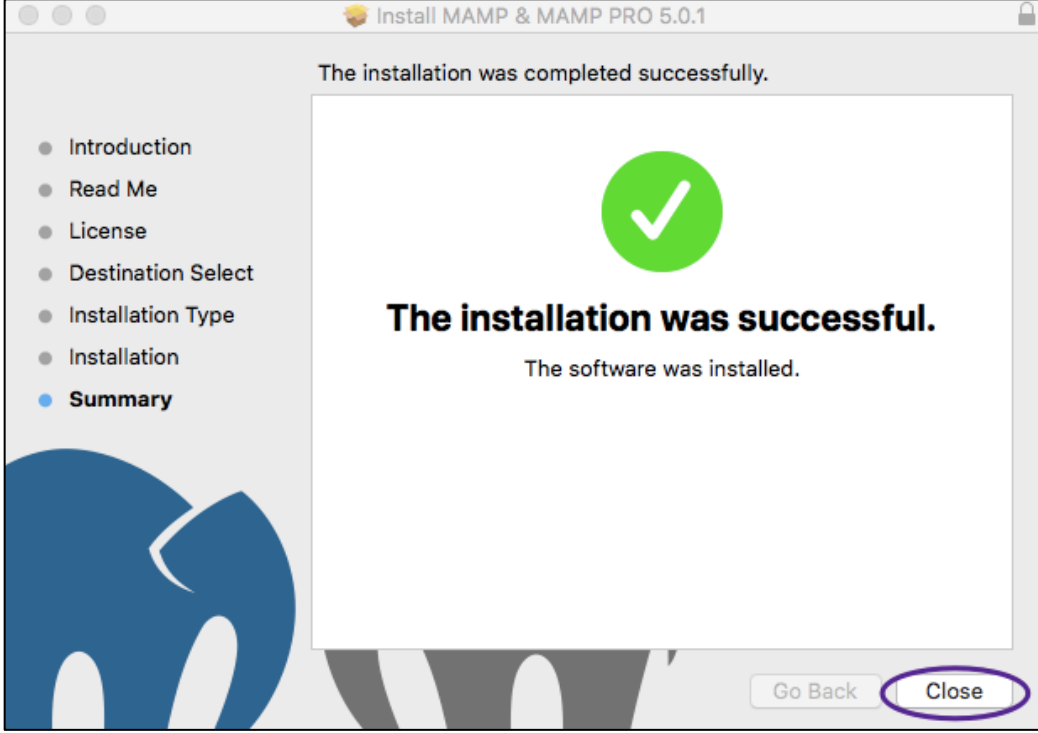
Resim 7.2.8

6. Son olarak açılan pencerede “**install (yükle)**” butonuna tıklayarak, indirme işlemini başlatabilirsiniz.



Resim 7.2.9

7. Açılan pencere “**close (kapat)**” butonuna basarak işlemi sonlandırınız. Uygulamanız çalıştırılmaya hazır olacaktır.



Resim 7.2.10

7.3 HTML TARİHİ

- **1989** yılında Tim Berners-Lee **WWW (World Wide Web)** i geliřtirdi.
- **1991** yılında Tim Berners-Lee **HTML**'i geliřtirdi.

NOT: HTML (Hyper Text Markup Language – Hiper Metin İřaret Dili) web sayfalarının nasıl iřleneceđini gsteren kodlardır. Browserlar istenen web sayfasını ncelikle kendileri indirir ve daha sonra bu sayfayı gsterir. HTML, bu sayfanın nasıl bir grnme sahip olacađını belirleyen kısımdır.

- **1993** yılında Dave Ragert **HTML+**'ı geliřtirdi.
- **1995** yılında **HTML 2.0** geliřtirildi.

NOT: 1994 yılında **W3C** kurulması ile birlikte HTML alıřmaları zerinde geliřtirici etkileri olmuřtur.

- **1997** yılında **W3C**, **HTML 3.2**'yi geliřtirmiřtir.
- **1999** yılında **W3C**, **HTML 4.01**'i geliřtirmiřtir.
- **2000** yılında **W3C**, **XHTML 1.0**'ı geliřtirmiřtir.
- **2008** yılında **WHATWG** tarafından **HTML5** geliřtirildi.
- **2012** yılında **WHATWG** tarafından **HTML5** hayata geirildi.
- **2014** yılında **W3C** tarafından **HTML5** geliřtirildi.
- **2016** yılında **W3C**, **HTML 5.1**'i geliřtirmiřtir.

7.4 HTML NEDİR?

HTML, bir sayfanın görselleştirilmesi için kendi içinde kodları bulunan fakat programlama dili olmayan bir işaret dilidir. Türkçe karşılığı Hiper Metin İşaretleme Dili'dir.

Görsel, metin ve video gibi içeriklerin web sayfasında yer alan konumlarını belirlemenize yardımcı olmaktadır.

Web sayfaları ile ilgili bilgileri arama motorlarına aktarmaktadır. Ayrıca oluşturduğunuz içeriklerin internet sitenizde düzgün bir şekilde görüntülenmesini sağlamaktadır.

HTML5 için geliştirilen logo Resim 7.4.1'de yer almaktadır.



Resim 7.4.1

7.4.1 HTML KODLARI HAKKINDA ÖNEMLİ NOKTALAR

- Açılan etiketlerin “/” (slash) işareti ile kapatılması gerekmektedir.
- Html kodları yazılırken Türkçe karakterlere yer verilmemektedir.
- Etiketler yazılırken büyük ya da küçük harfler arasında bir fark bulunmamakla birlikte, genellikle küçük harfler tercih edilmektedir.
- “<” ve “>” şeklinde büyüktür ve küçüktür ifadesi ile yazılan birime **tag** denilmektedir.

7.5 GENEL BAKIŞ

Html kodlarını yazarken, yazılacak olan her komut küçüktür ve büyüktür (< - >) arasına yazılmaktadır (örneğin <html>). Açılan komutların arasına yazılmasını istediğiniz nitelikleri belirttikten sonra, her kodu kapatmanız gerekmektedir. Kapatma işlemi slash işareti (/) ile konulmaktadır (örneğin </html>).

Bazı komutlar tek satırlıdır. Bunun anlamı o komutların açılma ve kapanma işlemini ayrı ayrı yapmalarıdır. Yani örneğin sayfaya bir çizgi çizmek istiyorsunuz. Bu işlemin bir başlangıç ve bitiş noktası olmalı ve ikisi de birlikte olduğu için slash işareti komutun sağ kısmına gelerek ifade edilmektedir (örneğin <hr/>).

Resim 7.5.1 de temel bir web sayfasının HTML kodlarının görünümü bulunmaktadır. Genel olarak her Html kodu <html> komutu ile başlar ve bu komutun kapatılmış hali ile biter (</html>).

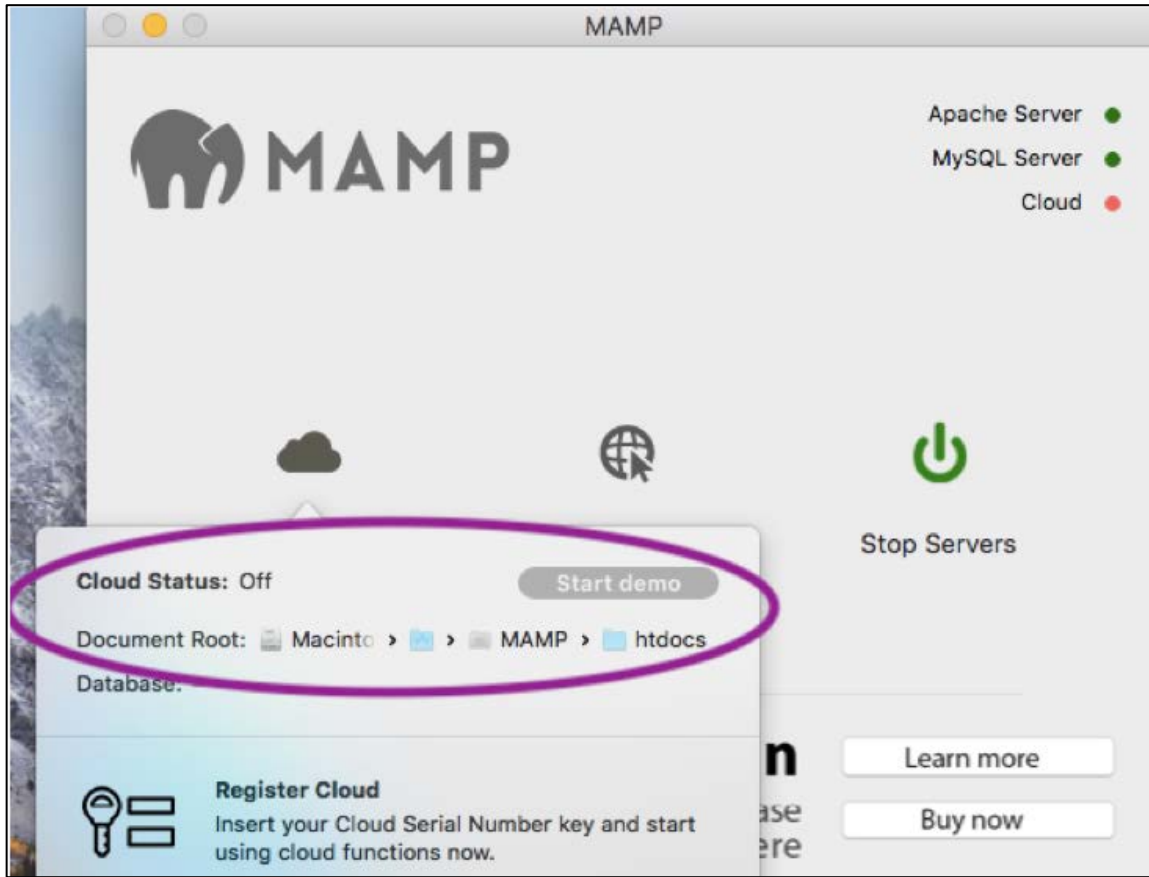
```
1 <html>
2
3
4 <head>
5
6 <title> bilgisayar kavramaları </ title>
7
8 </head>
9
10
11 <body>
12
13 <p> Html derslerine giriş </ p>
14
15 </ body>
16
17
18 </html>
19
20
```

Resim 8.5.1

Ayrıca Html kodları kendi aralarında iki bölümden oluşmaktadırlar. İlk bölüm **head** kısmıdır ve bu kısımda sayfa üzerinde gözükmeyen fakat sayfanın üst bilgilerini oluşturan komutlar yer almaktadır. Örneğin resimde gördüğümüz başlık yani **title** kısmı, sayfanın içeriğini değil, bir web sitesi açıldığı zaman, sekmenin ismini tanımlamaktadır.

İkinci kısmı ise gövde yani **body** kısmı bulunmaktadır. Sayfanın içeriğinin ve niteliğinin belirtildiği, kullanıcının sayfayı gördüğü halini oluşturan kısım burada yer almaktadır.

Açıklamaları uygulamaya dökmek gerekirse, ilk önce kurulumunu yaptığımız MAMP uygulamasını açıyoruz. Açılan ekranda **cloud** yazan kısma tıklayarak, işaretli kısımda belgenin nereye kayıt edileceğini kontrol etmelisiniz (Resim 7.5.2).



Resim 7.5.2

NOT: Tarayıcıların bir web sitesini açarken ilk olarak veriyi kendilerine indirdiklerini ve daha sonra Html kodlarını çalıştırdıklarından bahsetmiştik. Eğer isterseniz indirilen ve sayfanın arka kısmında çalışan Html kodlarını görebilirsiniz. Bunun için web sitesindeyken sağ tıklamanız ve **sayfa kaynağını görüntüle** demeniz gerekmektedir. Ya da tarayıcının ayarlar kısmında da bu seçenek yer almaktadır.

Uygulamamıza devam ederken, MAMP uygulamasında boş bir belge açarak denemeler yapmaya başlayabiliriz. Bu belge herhangi bir editör olabilir.

Aşağıda yer aldığı gibi kodlarımızı **body** kısmı içerisine yazarak kapatıyoruz (head kısmını şimdilik yazmanıza gerek yoktur). Örneğin biz kodumuzda 'Bilgisayar Kavramları' yazıyoruz ve yazdığımız dosyayı kayıt ediyoruz.

```
<html>

  <head>
<meta charset="UTF-8" />
  </head>

  <body>

    Bilgisayar Kavramları <br>

  </body>

</html>
```

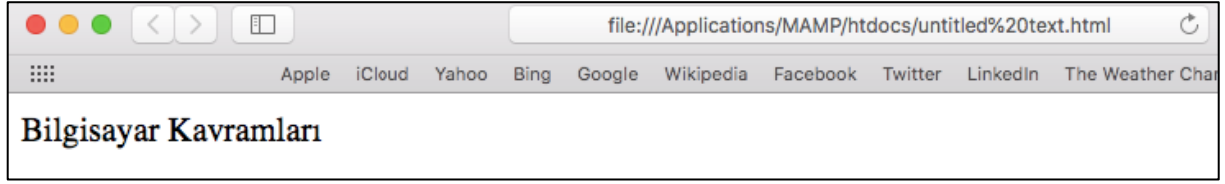
Resim 7.5.3

MAMP uygulaması içerisinde 'web sayfasını aç (open web page)' denildiği zaman, açılan sayfada URL kısmında <http://localhost:8888/MAMP/?language=English> gibi bir adres yer alacaktır. Sadece <http://localhost:8888> kısmını çalıştırmamız kodun kullandığımız browser üzerinde açılmasına yeterli olacaktır.

NOT: Kodlarınızı çalıştırdığınız zaman Türkçe karakter bulunan yerlerde sorun yaşıyorsanız, kodlarınızda **<head>** **</head>** kodları arasında, aşağıdaki kodları ekleyerek bu sorunu çözebilirsiniz.

```
<meta charset="UTF-8" />
```

Açılan sayfada görüntülenecek olan içerik Resim 7.5.4'deki gibidir:



Resim 7.5.4

NOT: Yapılacak olan uygulamaların çoğunu sadece tıklayarak yapmanızı sağlayan ve arka planda kendisi Html kodlarını yazan uygulamalar bulunmaktadır. Örneğin **Dreamweaver** gibi hazır bulunan bu uygulamaları bu kitapta kullanmayacağız çünkü Html bilmek için en azından genel olan bazı komutları bilmemiz gerekmektedir. **Notepad** ya da **Text Wrangler** gibi basit not araçlarını kullanmanın, komutları kendimiz yazdığımız için Html kodlarını öğrenebilmek açısından daha yararlı olduğunu düşünüyoruz.

NOT: Kod eklerken yazılarımızın yan yana ekli olarak gelmemesi için **</br> (break)** komutunu kullanabilirsiniz. Bu komut html kodları üzerinde, alt satıra geçilmesini sağlamaktadır. Örneğin siz kodlarınızın içerisine yazıları yazarken alt satıra geçerek yazsanız bile, komutları yazmadığınız takdirde, Html yazıları yan yana çalıştıracaktır. Dolayısıyla Html, komutları çalıştıran bir yapıya sahiptir.

7.6 ETİKETLER (TAGS)

Kodlarımızı yazarken kullandığımız etiketler kendi aralarında **fiziksel** ve **mantıksal** etiketler olarak ikiye ayrılmaktadırlar. Fiziksel etiketler, şu ana kadar da hali hazırda kullandığımız temel etiketleri içerirken, mantıksal etiketler daha çok detaylara ihtiyacımız olduğunda kullanılmaktadır.

Fiziksel etiketlere örnek olarak aşağıdakiler verilebilir:

- `<tt>`
- `<i>`
- ``
- `<u>`
- `<big>`
- `<small>`
- ``
- `<blink>`

Mantıksal etiketlere örnek olarak aşağıdakiler verilebilir:

- ``
- ``
- `<address>`
- `<blockquote>`
- `<acronym>`
- `<kbd>`

7.7 HYPERLINK

Hyperlink, bir metin veya görsel dosyaya eklenen, sizi yeni bir dosyaya veya dosyanın farklı bir bölümüne yönlendiren bağlantılardır.

Örnek olarak bilgisayarınızda kayıtlı olan bir resmi kolaylık olması için html dosyasının bulunduğu yere kayıt edebilirsiniz. Bu resmi web sayfanıza ekleyebilmek için **img (image)** komutunu kullanmanız gerekmektedir. İmg komutu içerisine resminizin ismini ve uzantısını yazmanız için **src (source – kaynak)** komutunu kullanmalısınız.

Örnek kodlarımız Resim 7.7.1 de verildiği gibi olmalıdır.

```
<html>

  <head>
  <meta charset="UTF-8" />
  </head>

  <body>

    Bilgisayar Kavramları <br>

    
  </body>

</html>
```

Resim 7.7.1

Çalıyor olduğunuz browser üzerinde sayfayı yenilediğinizde veya html kodlarınızı yazdığını html dosyanızı tekrar çalıştırdığınız zaman web sayfanızda, yüklemiş olduğunuz resim görülecektir.

Aynı zamanda mevcut resmin boyutları ile oynamak istersek genişlik anlamına gelen **width** ve yükseklik anlamına gelen **height** komutlarını kullanarak ayarlamamız mümkündür. Örneğin genişliği ve yüksekliği 400 px olacak şekilde resmi yayınlamasını istersek, örnek kodlarımız Resim 7.7.2'deki gibi olmalıdır.

Çalışıyor olduğunuz browser üzerinde sayfayı yenilediğinizde veya html kodlarınızı yazdığınız dosyayı tekrar çalıştırdığınız zaman web sayfanızda diğer resme göre boyutu daha farklı gözükecektir (orijinal boyuta göre daha büyük ya da küçük).

```
<html>

  <head>
<meta charset="UTF-8" />
  </head>

  <body>

    Bilgisayar Kavramlari <br>

    
  </body>

</html>
```

Resim 7.7.2

7.8 KAÇIŞ KODLARI (ESCAPE CODES)

Html sayfasında kodlarınızı yazarken, Html komut sembollerinden birini veya özel bir karakteri kullanmanız gerekiyorsa editörün, özel karakterleri algılayabilmesi için düzenlenen kodlara **kaçış kodları (escape codes)** denir. Bu kodlara örnek olarak “<”, “>”, “/”, “(”, “)” verilebilir.

Aşağıdaki buna benzer sembollerin bazılarının kod karşılıkları verilmiştir. İsterseniz internet üzerinden “**Html Symbols**” şeklinde aratarak tüm sembolere erişebilirsiniz.

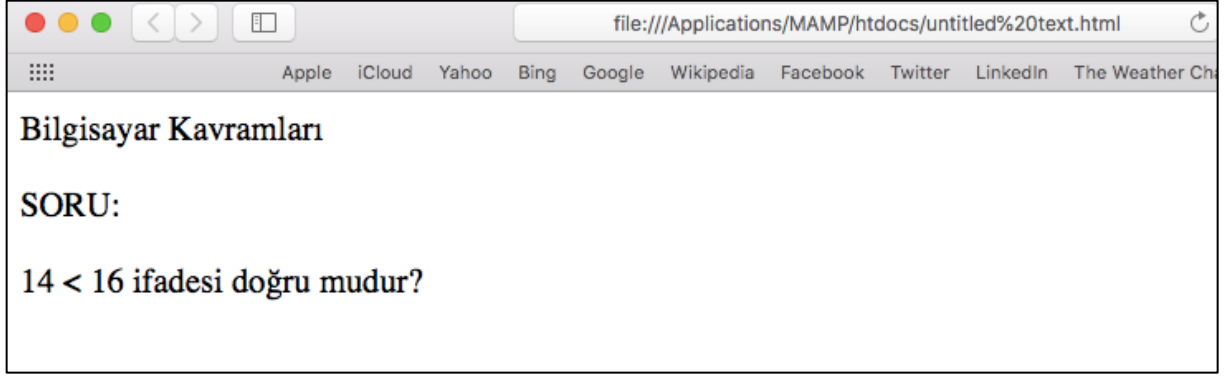
SEMBOLLER	KODLAR (SAYI)	KODLAR (STRING)
<	<	<
♥	♥	♥
←	←	←
€	€	€
Δ	Δ	Δ
∀	∀	∀
ε	∈	∈
®	®	®
♣	♣	♣
↓	↓	↓

Kaçış kodlarını kodlarımızda kullanabilmek için, kullanıyor olduğumuz editör üzerine kodların isminin ya da sayı versiyonunun yazılması yeterlidir. Örnek olarak, iki sayının birbirine göre büyüklüğünü kontrol edebilmek için **<** yazmamız gerekmektedir.

Örnek kodlarımız Resim 7.8.1 de ve uygulamamızı çalıştırdığımızda sonuç Resim 7.8.2 de verilmiştir.

```
1 <html>
2
3
4 <head>
5
6 <meta charset="UTF-8"/>
7
8 </head>
9
10
11 <body>
12
13 <p> Bilgisayar Kavramları </ p>
14
15 <br>
16 <br>
17 SORU:
18 <br>
19 <br>
20 14 &lt; 16 ifadesi doğru mudur?
21
22
23 </body>
24
25
26 </html>
```

Resim 7.8.1



Resim 7.8.2

7.9 HTML – LİSTELER

HTML kodlarını yazarken 3 farklı listeleme yapabilirsiniz.

- 1) **Düzenli liste (Order List)**
- 2) **Tanımlanmamış liste (Undefined List)**
- 3) **Tanımlanmış liste (Defined List)**

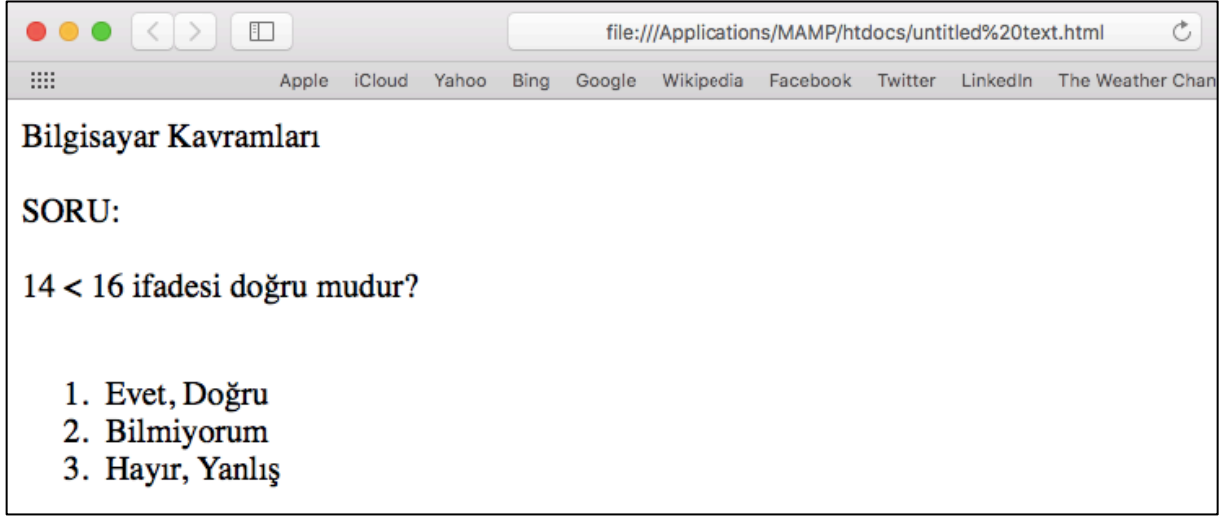
Editörümüzde bu kodları yazarken, kodlarımız baş harflerinden oluşmaktadır. Örneğin Order List yazmak isterseniz ****, Undefined List için ****, Defined List için **<dl>** yazmalısınız.

Bu kodlar, yazacağınız seçeneklerin biçimlerini belli etmektedir. Ayrıca liste elemanlarını yazmak için de **** ve **** elemanlarını kullanmalısınız.

Örnek bir liste yapmak için, daha önce yazmış olduğumuz sorunun seçeneklerini listeleme ile ekrana bastıralım. Bu kodları Order List ile yapmak istersek, kodlarımızın editördeki ve ekranda çalışmış hali Resim 7.9.1 ve Resim 7.9.2 de verilmiştir.

```
17 SORU:  
18 <br>  
19 <br>  
20 14 &lt; 16 ifadesi doğru mudur?  
21 <br>  
22 <br>  
23 <ol>  
24  
25 <li>Evet, Doğru</li>  
26 <li>Bilmiyorum</li>  
27 <li>Hayır, Yanlış</li>  
28  
29 </ol>
```

Resim 7.9.1



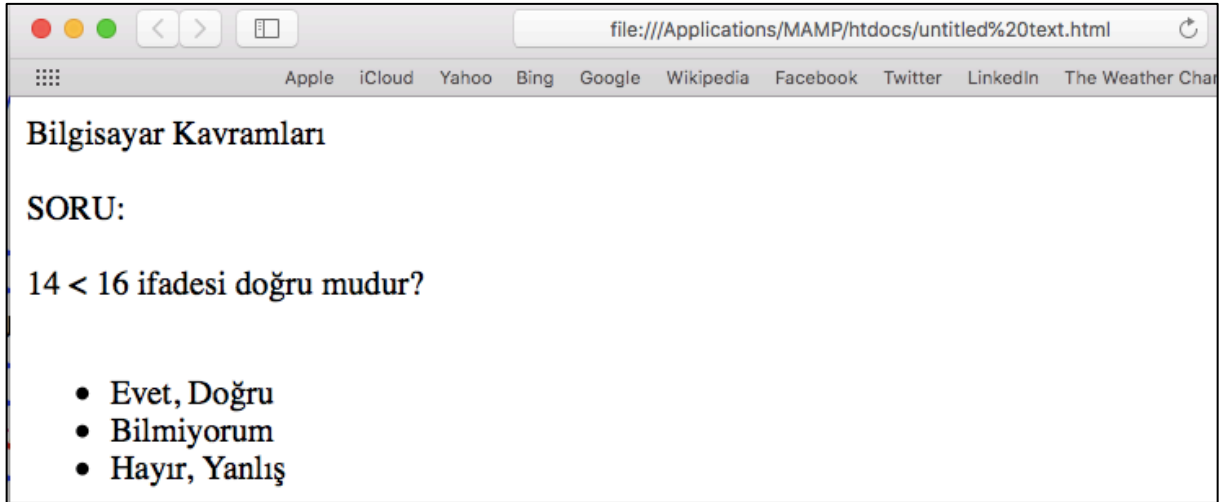
Resim 7.9.2

Görüldüğü üzere Düzenli Liste ile yapıldığı zaman, seçenekler sayısal sıralama ile birlikte gelmektedir.

Tanımlanmamış Liste ile yapmak istersek, kodlarımızın editördeki ve ekranda çalışmış hali Resim 7.9.3 ve Resim 7.9.4'deki olacaktır.

```
17     SORU:
18     <br>
19     <br>
20     14 &lt; 16 ifadesi doğru mudur?
21     <br>
22     <br>
23     <ul>
24
25     <li>Evet, Doğru</li>
26     <li>Bilmiyorum</li>
27     <li>Hayır, Yanlış</li>
28
29     </ul>
```

Resim 7.9.4



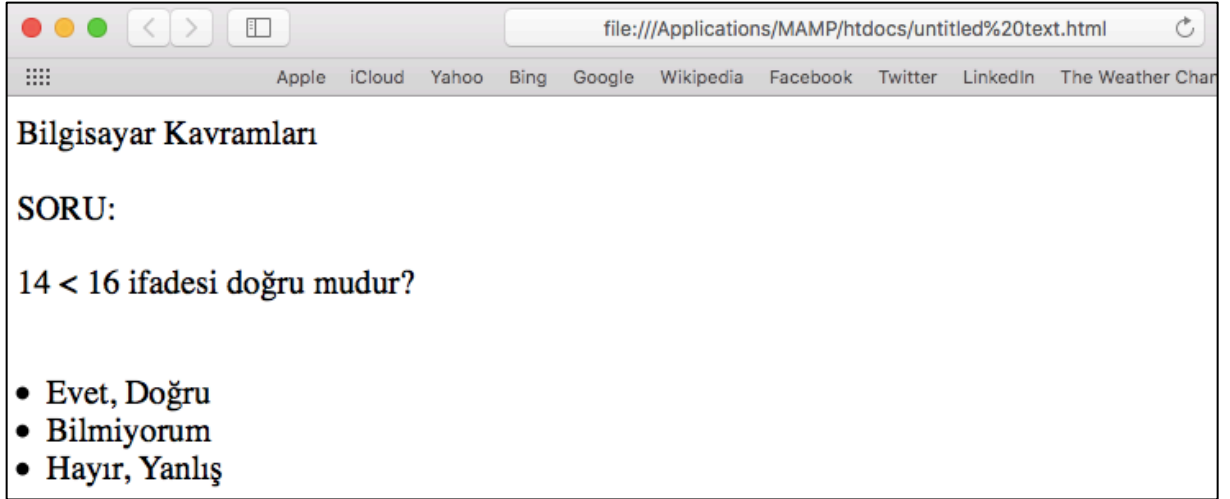
Resim 7.9.4

Aynı seçenekleri Tanımlanmamış Liste ile yaptığımız zaman sıralama sayılar ile değil, şekil ile belirtilmektedir.

Son olarak Tanımlanmış Liste ile yapmak isterse, kodlarımızın editördeki ve ekranda çalışmış hali Resim 7.9.5 ve Resim 7.9.6'dır.

```
17 SORU:
18 <br>
19 <br>
20 14 &lt; 16 ifadesi doğru mudur?
21 <br>
22 <br>
23 <dl>
24
25 <li>Evet, Doğru</li>
26 <li>Bilmiyorum</li>
27 <li>Hayır, Yanlış</li>
28
29 </dl>
```

Resim 7.9.5



Resim 7.9.6

Yazmış olduğumuz son listeleme yönteminin (Tanımlanmış Liste) gösterim şekli sayılar ile olmayıp, Tanımlanmamış Liste şekline benzer şekilde noktalar ile belirtilmektedir. Ancak Tanımlanmamış Liste yöntemine göre seçenekler paragrafın iç kısmına girerek değil, paragraf hizasından oluşturulmuştur.

7.10 HTML – TABLOLAR

Html ilk kodlanmaya başlandığı zamanlarda, sadece tablo oluşturmak için kullanılmamaktaydı. İki resmi yan yana koymak gibi işlemler için de tablo kodlarından yararlanılmaktaydı.

Şimdilerde oluşturacak olduğumuz Html tablosu için ana kodumuz **<table>** ve **</table>** ‘dır. Daha sonra kodlarımızı yazarken, satır satır düzenleyerek yazmamız gerekmektedir. Yani önce 1. Satırdakiler, ardından 2., 3. şeklinde ilerlemektedir. Satırları oluşturmak için **<tr>** ve **</tr>** kodunu kullanmaktayız ve o satırın sütunlarını da içerisine yazdığımız **<td>** ve **</td>** kodu ile sağlamaktayız.

NOT: Html kodlarını yazarken yorum satırı oluşturabilmek için, **<!--** ve **-->** şeklinde belirtebiliriz. İki kod arasına yazdığımız şeyler geçerli sayılmayacak ve çalıştırılmayacaktır.

Dersin adının, dersin kodunun ve kredisinin yazıldığı bir örnek tablo oluşturmak istersek, kodlarımız Resim 7.10.1 ve ekrana yansıyan çıktısı ise 7.10.2'deki gibi olacaktır.

```
<table>

<tr>

<td> Dersin İsmi </td>
<td> Dersin Kodu </td>
<td> Dersin Kredisi </td>
</tr>

<tr>
<td> Programlama </td>
<td> CSC 04 </td>
<td> 3 </td>
</tr>

<tr>
<td> Makine Öğrenmesi </td>
<td> CSC 05 </td>
<td> 4 </td>
</tr>
```

Resim 7.10.1

Bilgisayar Kavramları		
Dersin İsmi	Dersin Kodu	Dersin Kredisi
Programlama	CSC 04	3
Makine Öğrenmesi	CSC 05	4

Resim 7.10.2

Yazmış olduğumuz tabloda ekranda görüldüğü üzere satırlar ve sütunlar çizgiler ile ayrılmamıştır. Bu <table> kodunun bir özelliği olduğu için, içerisine **border=1** şeklinde belirtmeniz gerekmektedir. Yazmış olduğumuz sayı kenarın kalınlığını belirlediğinden, ne kadar sayıyı artırırsanız, o kadar kalın bir çizgiler elde edersiniz.

```
<table border=1 >

<tr>

<td> Dersin İsmi </td>
<td> Dersin Kodu </td>
<td> Dersin Kredisi </td>
</tr>

<tr>
<td> Programlama </td>
<td> CSC 04 </td>
<td> 3 </td>
</tr>

<tr>
<td> Makine Öğrenmesi </td>
<td> CSC 05 </td>
<td> 4 </td>
</tr>
```

Resim 7.10.3

Bilgisayar Kavramları		
Dersin İsmi	Dersin Kodu	Dersin Kredisi
Programlama	CSC 04	3
Makine Öğrenmesi	CSC 05	4

Resim 7.10.4

Aynı zamanda tablo oluşturduktan sonra birçok özellik eklemeniz mümkündür. Örneğin yazdığınız tabloda kolonları ya da satırları da birleştirmeniz mümkündür. Bunun için kolonları birleştiren **colspan (column span)**, satırları birleştiren **rowspan** olarak bilinmektedir.

Daha iyi açıklayabilmek adına rowspan elementinden yola çıkarak bir örnek yapalım. Bunun için örneğimize öncelikle daha fazla satır ekliyoruz. Hangi satırdan itibaren ve kaç tane satırın birleştirilmesini istiyorsak, oraya rowspan kodunu yazıyoruz (örneğin rowspan=2 şeklinde).

```
<table border=1 >

<tr>

<td> Dersin İsmi </td>
<td> Dersin Kodu </td>
<td> Dersin Kredisi </td>
</tr>

<tr>
<td> Makine Öğrenmesi </td>
<td> CSC 04 </td>
<td> 3 </td>
</tr>

<tr>
<td> Veri Bilimi </td>
<td> CSC 05 </td>
<td> 3 </td>
</tr>

<tr>
<td rowspan=2> Seçmeli Dersler </td>
<td> CSC 07 </td>
<td> 3 </td>
</tr>

<tr>
<td> CSC 08</td>
<td> 2 </td>
</tr>
```

Resim 7.10.5

Bilgisayar Kavramları

Dersin İsmi	Dersin Kodu	Dersin Kredisi
Makine Öğrenmesi	CSC 04	3
Veri Bilimi	CSC 05	3
Seçmeli Dersler	CSC 07	3
	CSC 08	2

Resim 7.10.6

Kolonları birleřtirmek için kullandığımız colspan için de yapacağımız örnekte, en üst kısma bir satır daha ekleyelim. Ve ‘Dersin Kodu’ kısmından sonra gelen ‘Dersin İsmi’ ve ‘Kredisi’ kısmını, “detaylar” başlığı altında toplayalım.

```
<table border=1 >
<tr>
<td> </td>
<td colspan=2> Detaylar </td>

<tr>
<td> Dersin İsmi </td>
<td> Dersin Kodu </td>
<td> Dersin Kredisi </td>
</tr>

<tr>
<td> Makine Öğrenmesi </td>
<td> CSC 04 </td>
<td> 3 </td>
</tr>

<tr>
<td> Veri Bilimi </td>
<td> CSC 05 </td>
<td> 3 </td>
</tr>

<tr>
<td rowspan=2> Seçmeli Dersler </td>
<td> CSC 07 </td>
<td> 3 </td>
</tr>

<tr>
<td> CSC 08</td>
<td> 2 </td>
</tr>

</table>
```

Resim 7.10.7

Bilgisayar Kavramları

	Detaylar	
Dersin İsmi	Dersin Kodu	Dersin Kredisi
Makine Öğrenmesi	CSC 04	3
Veri Bilimi	CSC 05	3
Seçmeli Dersler	CSC 07	3
	CSC 08	2

Resim 7.10.8

Html üzerinde tablo yaparken aynı zamanda tablonuzun istediğiniz bir kısmına başlık ekleyebilmeniz de mümkündür. Bu işlevi `<th>` ve `</th>` kodu ile sağlayabilirsiniz. Böylece içerisine yazmış olduğunuz yazılar kalın bir şekilde görülecektir. Resim 7.10.9 da üst satıra eklediğimiz kodlarımız ve Resim 7.10.10.ekranda çalıştırılmış hali mevcuttur.

```
<table border=1 >

<tr>
<th colspan=3> Dersler Hakkında </th>
</tr>

<tr>
<td> </td>
<td colspan=2> Detaylar </td>
</tr>

<tr>
<td> Dersin İsmi </td>
<td> Dersin Kodu </td>
<td> Dersin Kredisi </td>
</tr>
```

Resim 7.10.9

Bilgisayar Kavramları

Dersler Hakkında		
	Detaylar	
Dersin İsmi	Dersin Kodu	Dersin Kredisi
Makine Öğrenmesi	CSC 04	3
Veri Bilimi	CSC 05	3
Seçmeli Dersler	CSC 07	3
	CSC 08	2

Resim 7.10.10

Html etiketleri üzerinde birkaç alıştırma yapmak için, ekrana yazdığımız bir cümleyi italik, kalın, altı çizgili şekilde nasıl yazılacağı resimler ile gösterilecektir. Daha iyi anlaşılması açısından ilk örneklerde etiketleri ayrı ayrı yazacağız. Son cümlede de görüleceği üzere, bu özellikler aynı cümle içerisinde ya da hepsi aynı anda kullanılabilir.

Resim 7.10.11 de görüldüğü üzere, cümleyi kalın yazmak için **** ve ****; italik yazmak için **<i>** ve **</i>**; altı çizili yapmak için **<u>** ve **</u>** elementlerinden yararlanılmaktadır.

```
<b> "İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları </b>
<br>
<br>
<br>
<i> "İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları </i>
<br>
<br>
<br>
<u> "İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları </u>
<br>
<br>
<br>
<b> "İnternet Nasıl Çalışır" <b> <i> videosu </i> <u> etiketler </u> konusu <b> <i> <u> alıştırma çalışmaları </b> </i> </u>
```

Resim 7.10.11

Kodlarımızı çalıştırdığımız zaman Resim 7.10.12 de görüldüğü üzere ilk cümle kalın, ikinci cümle italik, üçüncü cümle altı çizili yazılmıştır. En son cümle ise hepsinin ayrı ayrı bir cümlede veya aynı cümlenin hepsine sahip olabileceğini göstermek amacı ile yazılmıştır.

Bilgisayar Kavramları

"İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları

"İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları

"İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları

"İnternet Nasıl Çalışır" videosu etiketler konusu alıştırma çalışmaları

Resim7.10.12

Az önce bahsedilen etiketler Html'in temel özelliklerinden birkaçıdır. Aynı özellikleri daha da geliştirebilmek için **** ve **** olarak verilen bir diğer etiketi kullanabilirsiniz. Bu etiket ile birlikte yazının boyutunu, rengini veya arka plan rengi gibi özelliklerini ayarlayabilmeniz mümkündür.

Yazının boyutunu ayarlamak için **size**, rengini ayarlamak için de **color** özelliğinden yararlanacağız.

Örnek olarak yazdığımız cümlede boyutunu 10 ve rengini de pembe olarak belirliyoruz.

```
<font size=10; color=pink > İnternet Nasıl Çalışır Videosu Geliştirilmiş Etiketler </font>
```

Resim 7.10.13

Kodlarımızın çalıştırılmış hali Resim 7.10.14'te verilmiştir.

İnternet Nasıl Çalışır Videosu Geliştirilmiş Etiketler

Resim 7.10.14

Renkleri yazarken belirli bir kelime kapasitesi olduđu için, color etiketini bu şekilde kullanmak size kısıtlama sağlayacaktır. Bu gibi özellikleri daha gelişmiş olarak kullanabilmeniz için rengin kodunu bularak da yazabilirsiniz. Bunun için arama motorunuza “color picker” yazarsanız eğer, renk skalasında işaretlediğiniz yerin kodunu verecektir (örneğin #6779d3 gibi).

Aynı zamanda **style** etiketi içerisinde, rengini belirttiğiniz alanın genişlik ve yüksekliğini de ayarlayabilmekteyiz. Bunun için **height** ve **width** kodları içerisine 8in (inç) şeklinde belirtebilirsiniz.

Örnek kodlarımız Resim 7.10.15’te ve sonuçların çıktıları 7.10.16’da verilmiştir.

```
<tr>
<th colspan=3 style= " height:3in; width:5in; " > Dersler Hakkında </th>
</tr>

<tr>
<td> </td>
<td colspan=2> Detaylar </td>
</tr>
```

Resim 7.10.15

Bilgisayar Kavramları

Dersler Hakkında

	Detaylar	
Dersin İsmi	Dersin Kodu	Dersin Kredisi
Makine Öğrenmesi	CSC 04	3
Veri Bilimi	CSC 05	3
Seçmeli Dersler	CSC 07	3
	CSC 08	2

Resim 7.10.16

Font etiketi ile aynı görevleri yerine getirebilen bir diğer etiket ise **<style>** etiketidir. Örneğin yazmış olduğumuz tabloda belirli bir kısmı, bu etiketin yardımı ile renk, boyut, arka plan gibi özellikler belirterek değiştirebilmemiz mümkündür.

Eğer yazının rengini değiştirmek istiyorsak **color**, yazının boyutunu değiştirmek istiyorsak **font-size** veya arka plan rengini değiştirmek istiyorsak **background-color** etiketlerini kullanarak bu değişiklikleri yapabilmemiz mümkündür.

```
<tr>
<td style= "background-color: blue; color: white; font-size:30; "> Dersin İsmi </td>
<td> Dersin Kodu </td>
<td> Dersin Kredisi </td>
</tr>
```

Resim 7.10.17

Örnekte görüldüğü üzere daha önceden oluşturmuş olduğumuz tablomuzda “Dersin İsmi” kısmının rengini, boyutunu ve arka plan rengini değiştirmiş bulunmaktayız. Kodlarımızı çalıştırdığımız zaman çıktıları Resim 7.10.18’deki gibi görünecektir.

Bilgisayar Kavramları		
Dersler Hakkında		
	Detaylar	
Dersin İsmi	Dersin Kodu	Dersin Kredisi
Makine Öğrenmesi	CSC 04	3
Veri Bilimi	CSC 05	3
Seçmeli Dersler	CSC 07	3
	CSC 08	2

Resim 7.10.18

7. 11 RESİM FORMATLARI

JPEG/JPG

Gerçek renk değerini içeren bir sıkıştırma programıdır. 256 ve daha fazla renk kullanan resimler için idealdir. Dolayısıyla az renk içeren resimlerin kalitesini düşürmektedir. Ayrıca megabyte olan resim boyutlarını kolaylıkla kilobyte olarak düzenleyebilme özelliğine sahiptir.

PNG

Kayıpsız bir şekilde resmi saklamak için kullanılan saklama biçimidir. GIF resim formatının gelişmiş formatı olarak düşünülebilir. Ancak hareketli resimleri desteklememektedir. Çizim ve metinlerde JPEG'den daha iyi olduğu söylenebilir. Aynı zamanda TIFF resim formatına göre de daha yaygın kullanılmaktadır.

BMP

En temel renk formatıdır. BMP, 16 ya da daha çok renk kaydedeceğinizi herhangi bir sıkıştırma yapmayan oldukça hızlı bir formattır. Resimdeki renk sayısından çok, resmin büyüklüğü önemlidir.

TIFF

Yüksek renk derinliği olan resimlerde kullanılır. Dosya boyutları JPEG'e göre daha büyüktür. TIFF biçimi birden fazla sayfayı destekleyebilmektedir. Dolayısıyla çok sayfalı tek bir TIFF dosyası olarak kaydedilebilmektedir.

8.BÖLÜM: CSS – STİLLER (STYLES)

8.1 CSS – STİLLER (STYLES)

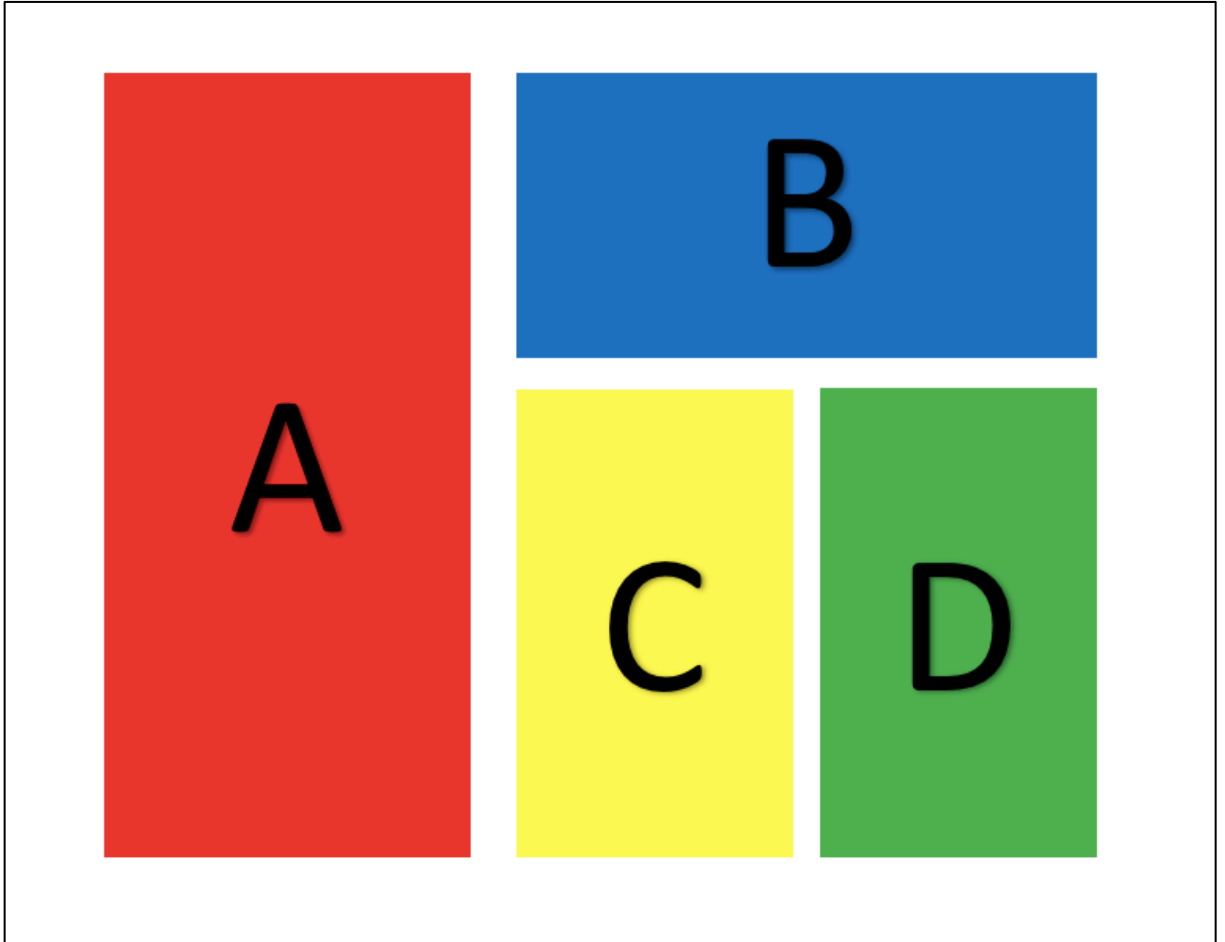
CSS üzerinde sayfanıza görsellik getirerek geliştirebilmenizi sağlayan 3 farklı stil verme yöntemi mevcuttur. Bunlardan ilki bir **inline** olarak yani bir liste elemanının yanında, o kodun içerisinde belirtilen style elementidir.

İkinci yöntem, halihazırda bulunan **<style>** etiketini açarak, yapmak istediğiniz görselleştirme çalışmalarını bu kodun içerisinde belirtmektir.

Son yöntemi ise Html kodlarımızı yazdığımız sayfa haricinde, bir CSS sayfası oluşturarak, o sayfayı Html sayfamıza entegre ederek gerçekleştirmektir.

Bu bölümün devamında her yöntem gösterilecektir fakat ilk olarak, inline olarak görselleştirme yöntemini deneyeceğiz.

Hem örneklerle aklınızda daha kalıcı hale gelmesini sağlamak, hem de tablolar konusu ile entegre olmasını sağlamak için, Resim 8.1.1’de görüldüğü gibi bir görseli, halihazırda öğrenmiş olduğumuz Html ve Css kodları ile yapacağız.



Resim 8.1.1

İlk olarak Resim 8.1.1'i inceleyecek olursak, görselde 3 kolon ve 2 satır bulunmaktadır. B bloğunda, iki sütun (colspan kodu kullanılmıştır) birleştirildiği için tek sütun görünümündedir.

Ayrıca A bloğunda da iki satır birleştirildiği için (rowspan kodu kullanılmıştır), tek satır görünümündedir.

İlk olarak temel Html kodlarımızı yazıyoruz (<html>, </html>, <head>, </head>, <body>, </body>).

Daha sonra <tr> ve </tr> etiketleri ile ilk satırımızı açıyoruz ve içerisine iki sütundan oluştuğu için iki kere <td> ve </td> kodumuzu ekliyoruz.

İlk oluşturduğumuz sütun etiketimizin içerisine A bloğunu koyacağımız için, yazı kısmına 'A' ve iki satırı birleştirdiği için rowspan='2' şeklinde ekliyoruz.

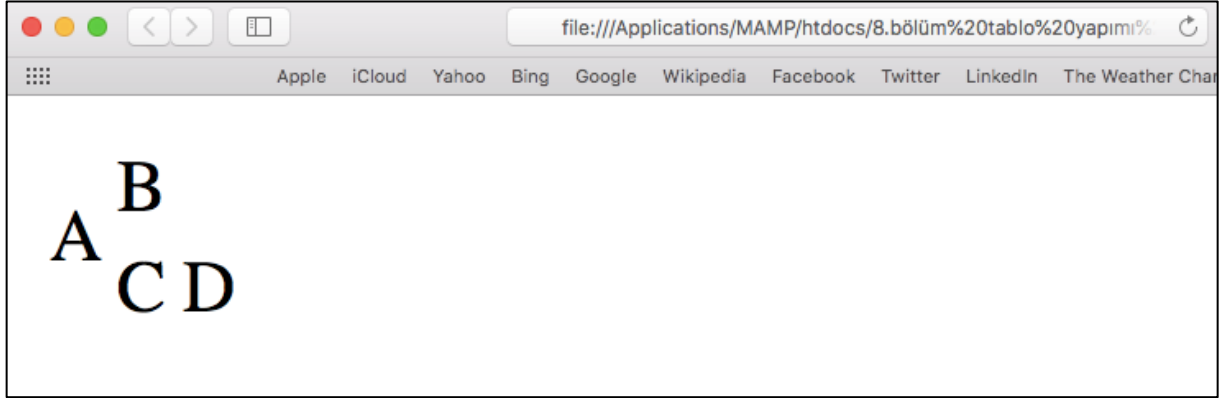
Aynı işlemi ikinci oluşturduğumuz sütun etiketimizin içerisine yazı kısmına 'B' ve colspan='2' şeklinde ekleyerek bitiriyoruz.

İlk satırımızı düzenledikten sonra ikinci satırımız için bir tane daha <tr> ve </tr> etiketi ekliyoruz. Oluşturacak olduğumuz 'B' ve 'C' blokları için <td> ve </td> etiketlerini yazıyoruz.

Kodlarımız Resim 8.1.2 de ve tarayıcımızda çalıştırdığımız zaman çıktılar Resim 8.1.3 verilmiştir.

```
11 <table>
12
13 <tr>
14 <td rowspan='2' > A </td>
15 <td colspan='2'> B </td>
16 </tr>
17
18 <tr>
19 <td>C</td>
20 <td>D</td>
21 </tr>
22
23 </table>
```

Resim 8.1.2



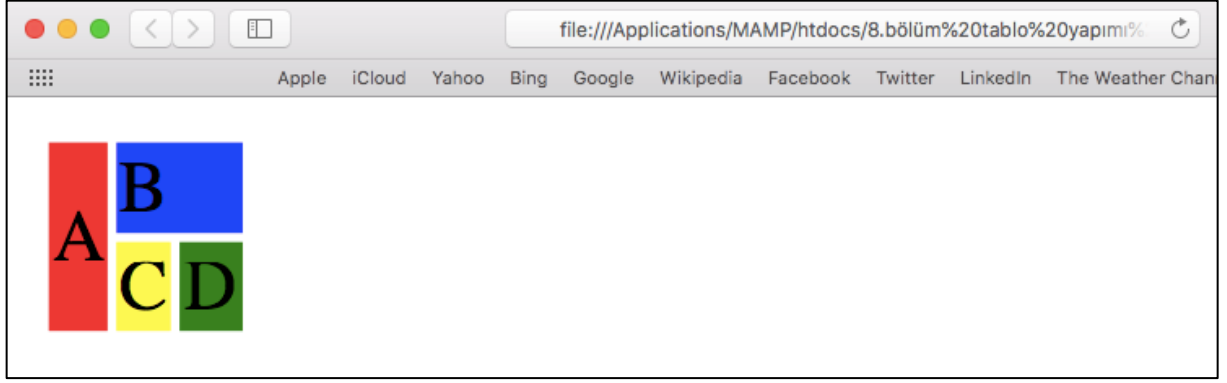
Resim 8.1.3

Resim 8.1.3'te görüldüğü üzere tablomuz ana hatları ile oluşturulmuştur. Arka planına renk vermek için Css'in style etiketinden yararlanacağız.

D bloğu için **style="background-color:green;"**, C bloğu için **style="background-color:yellow;"**, B bloğu için **style="background-color:blue;"** ve son olarak A bloğu için **style="background-color:red;"** şeklinde renklerine göre düzenliyoruz. Eklemiş olduğumuz style etiketi ile birlikte kodlarımız Resim 8.1.4'te ve kodlarımızın çalıştırılmış halde çıktısı da Resim 8.1.5'te verilmiştir.

```
11 <table>
12
13 <tr>
14 <td rowspan='2' style="background-color:red;"> A </td>
15 <td colspan='2' style="background-color:blue;"> B </td>
16 </tr>
17
18 <tr>
19 <td style="background-color:yellow;" >C</td>
20 <td style="background-color:green;" >D</td>
21 </tr>
22
23 </table>
```

Resim 8.1.4



Resim 8.1.5

Resim 8.1.1 üzerinde görülen resmimiz, yukarıda mevcut olarak bulunmaktadır. Eğer boyutunu büyütmek istersek, style etiketinin içerisine (Birinci yöntem: inline yöntemi ile) `font-size:60` ekleyerek gerçekleştirebilirsiniz (sayı değişken olabilir).

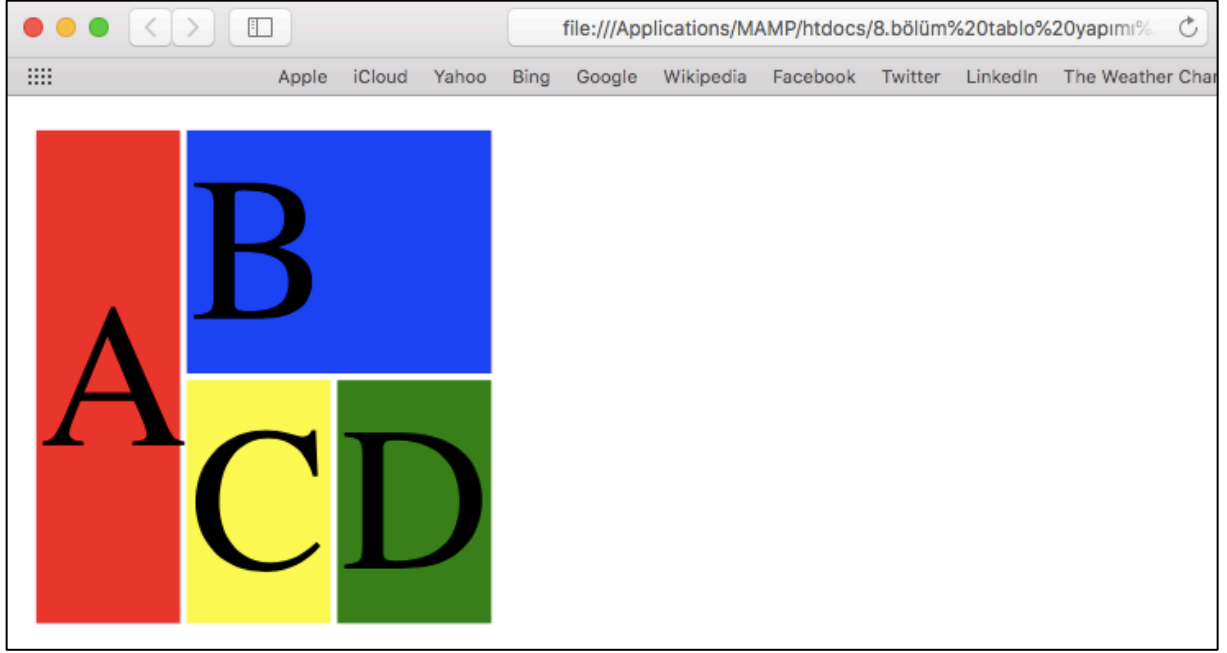
Boyutlandırma yaparken size daha farklı yöntemler gösterebilmek açısından, CSS ile style özelliği tanımlayabileceğimiz ikinci yolumuzu kullanacağız. İkinci yöntemimizde tablo için oluşturulan kodlarımızın içerisine değil, etiket olarak bir style kodu eklememiz gerekmektedir. Daha sonra kendi içerisinde hangi bölümde değişiklik yapacağımızı (td) belirtiyoruz ve yapacağımız değişikliği (`font-size:50`) yazıyoruz.

İkinci yöntemi kullanırken dikkat edilmesi gereken durum, açılacak style etiketinin <head> ve </head> etiketleri arasında yer alması gerektiğidir.

Kodlarımızın son hali Resim 8.1.6'da ve tarayıcımızda çalıştırılmış hali de Resim 8.1.7'de mevcuttur.

```
1 <html>
2
3 <head>
4
5 <meta charset="UTF-8"/>
6
7 <style>
8
9 td {
10 font-size:60;
11 }
12
13 </style>
14
15 </head>
16
17 <body>
18
19 <table>
20
21 <tr>
22 <td rowspan='2' style="background-color:red;"> A </td>
23 <td colspan='2' style="background-color:blue;"> B </td>
24 </tr>
25
26 <tr>
27 <td style="background-color:yellow;" >C</td>
28 <td style="background-color:green;" >D</td>
29 </tr>
30
31 </table>
32
33 </body>
34
35
36 </html>
```

Resim 8.1.6



Resim 8.1.7

Css ile ilgili diğer stil özelliklerine erişmek isterseniz, tarayıcınıza “**Css Commands**” yazarak, çıkan sayfalardan erişebilirsiniz.

NOT: Css ile ilgili diğer stil özelliklerine erişmek isterseniz, tarayıcınıza “**Css Commands**” yazarak, çıkan sayfalardan erişebilirsiniz.

Css ile stil özelliğini sadece metinlere, resimlere ya da tablolara değil, linklere de atayabilmekteyiz. Örneğin www.sadievrenseker.com adresini html dosyamıza eklemek için **a** etiketi altında, href kodu (hangi web sitesine bağlanmasını istiyorsanız) içerisine belirtiyoruz (<**a href="www.sadievrenseker.com"**>).

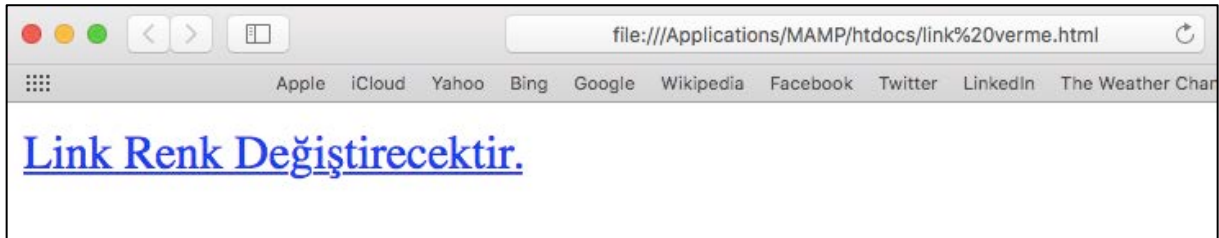
Örnek olarak bağlantısını verdiğimiz web sayfasının üzerine gelince renk değiştirmesini yapalım. Bunun için style kodları içerisinde **hover** kodunu kullanmamız gerekmektedir.

NOT: **Hover** özelliği bir arka planın, linkin ya da bir metnin renginin, gölgesinin ya da fontlarının değişmesini sağlamaktadır.

Kodlarımız Resim 8.1.8’de ve çalıştırılmış hali Resim 8.1.9 ve Resim 8.1.10’da bulunmaktadır.

```
3 <head>
4 <meta charset="UTF-8"/>
5
6 <style>
7
8 a:hover {
9
10 color:red;
11
12 }
13 </style>
14
15
16 </head>
17
18 <body>
19
20 <a href="www.sadievrenseker.com"> Link Renk Deęiřtirecektir. </a>
21
22 </body>
```

Resim 8.1.8



Resim 8.1.9

Üzerine gelinmeden önce link mavi iken, gelindikten sonra kırmızı olmaktadır.



Resim 8.1.10

Css üzerinden Html dosyamıza stil eklemek için üçüncü kullanacağımız yöntem ise, Html kodlarımızı yazdığımız dosyamız haricinde, bir de ayrıca Css dosyası oluşturarak, Html dosyamızla entegre etmektir.

Üçüncü yöntemin amacı herhangi bir yanlışlık olduğu zaman ya da kodlarınıza geri dönüp bir özelliği değiştirmek istediğiniz zaman, teker teker Html kodlarınızdan <style> kodunu değiştirmek yerine, Css dosyası üzerinden toplu olarak özelliği değiştirebilmeniz mümkündür.

Bunun için tek yapmanız gereken .css uzantılı bir dosya oluşturarak, Html dosyanızın <head> ve </head> kodları arasına **<link rel="stylesheet" type="text/css" href="dosyaadı.css" />** şeklinde oluşturmuş olduğunuz css dosyanızı belirtmenizdir.

Bu aşamadan sonra Html dosyanızda herhangi bir stil belirten kod olmaksızın, Css dosyasına yazdığımız kodlar ile entegre olacaktır.

Örnek olabilmesi açısından yukarıda göstermiş olduğumuz hover özelliğini bir css dosyasına aktarıyoruz ve kodlarımızı tekrar çalıştırıyoruz.

Html dosyasını görünümü Resim 8.1.11’de verilmiştir.

```
/Applications/MAMP/htdocs/link verme.html
1 <html>
2
3 <head>
4
5 <meta charset="UTF-8"/>
6 <link rel="stylesheet" type="text/css" href="dosyaadi.css" />
7
8 </head>
9
10 <body>
11
12 <a href="www.sadievrenseker.com"> Link Renk Değiştirecektir. </a>
13
14 </body>
15
16 </html>
```

Resim 8.1.11

```
/Applications/MAMP/htdocs/dosyaadi.css
1 <style>
2
3 a:hover {
4
5 color:red;
6
7 }
8 </style>
```

Resim 8.1.12

Kodların çalıştırılmış hali, Resim 8.1.10 ile aynı olacaktır.

9.BÖLÜM: HTML – FORMLAR

9.1 FORM NEDİR?

9.2 FORM NASIL OLUŞTURULUR?

Birçok yerde içerisinde yazı yazabildiğimiz metin kutucuklarına, tıklayabildiğimiz butonlar veya açılır kapanır menülere rastlamışsınızdır. Görmüş olduğunuz araçların çoğu, Html’de form özelliğinden yola çıkılarak yapılmış form bileşenleridir.

Bu özellikler herhangi bir yere giriş yaparken, kayıt olurken ya da bir sipariş verirken kullanılabilirler. Bu formlar aslında bizim ara yüz üzerinde kullandığımız ve arkasında bir yazılımın çalıştığı bileşenlerdir. Formların arkasında çalışan bu yazılımlar, günümüzde sunucu tarafında çalışan yazılımlar (**Server Side Scripting**) olarak geçmektedir. Doldurduğumuz formlar, onay butonuna bastığımız zaman, bu yazılımlar üzerinden gönderilmektedirler.

Bu bölümde formların ara yüz işlemlerinden, bir formun nasıl oluşturulduğu ve forma bilgilerin nasıl kayıt edildiği konularından bahsedilecektir.

9.1 FORM NEDİR?

Web sayfaları ya da tarayıcıların üzerinde çoğu zaman gördüğünüz **arama butonları (Search Box – Text Input)**, **tıklanabilir düğmeler (Buttons)**, **işaretlebilir kutular (Check boxes)**, **birden fazla seçim yapabilmemiz için kullanılan seçenekler (Radio Buttons)**, **tıklandığında açılabilir olarak geliştirilen menüler (Popup Menus)** form bileşenleridir.

9.2 FORM NASIL OLUŖTURULUR?

Formlar oluŖturulurken, bileŖenler **<form>** ve **</form>** kodları arasında oluŖturulurlar.

Her bir bileŖen oluŖturulurken diđer Html kodlarında olduđu gibi < ile baŖlayan ve > ile biten, tek satırlık kodlardan oluŖmaktadır. Diđer kodlardan ayrılan tek özelliđi kodu kapatmak zorunda olmamamızdır.

Form bileŖenlerinin nasıl oluŖtuđunu gsterirken, ncelikle bir metin kutusunun (Text Box) ve iŖaretlenebilir kutuların (check box) nasıl oluŖtuđunu gstereceđiz. Bunun iin her iki form bileŖeninin de zelliklerini **input** kodu ierisinde belirteceđiz.

Input etiketi zerinde yer alan zelliklerden **type** (tr) hangi form bileŖeni olduđunu belirlemektedir. rneđin metin kutusu yazmak istiyorsak **text**, ierisine tıklanıđında tık oluŖmasını sađlayan iŖaretleme kutucuđu yapmak istiyorsak **checkbox** yazmanız gerekmektedir.

Diđer bir verilen zellik **name** (isim)'dir. zellikle ok fazla form bileŖeni olduđu zaman kodlarımız arasından ayırt edilebilmesini sađlamak amalı her birine zel isim vermemiz gerekmektedir.

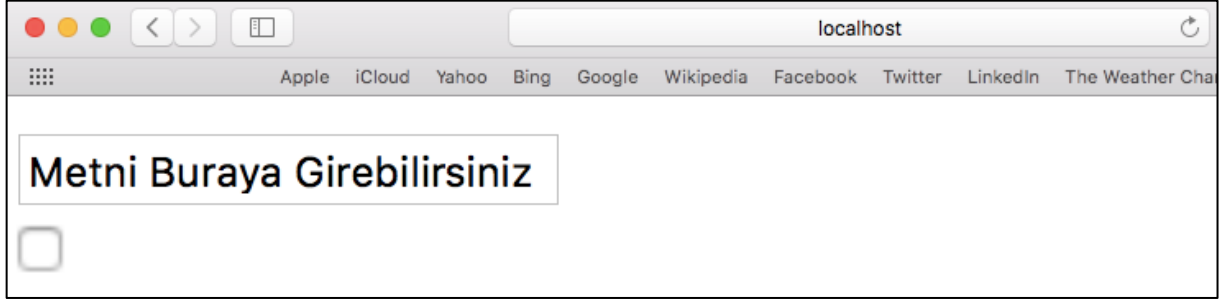
Son verdiđimiz zellik ise **value** (deđer) zelliđidir. Eđer yazdıđımız form bileŖeninin, ierisinde bir bilgi ile gelmesini istiyorsanız, bilgiyi value kısmına girmeniz gerekmektedir. rneđin yapılan arama kutularının ierisinde 'site ii arama' gibi ifadeler yer aldıđını daha nceden grmŖsnzdr. Bu gibi ifadeler iin form bileŖeninin value zelliđi kullanılmaktadır.

Value zelliđini atarken aynı durum check box iin farklıdır. IŖaretleme yapılan kutucukların ierisine herhangi bir bilgi koyulamayacađı iin sadece 'yes' ya da 'no' olarak belirtebilmekteyiz. Bu da onların, kodlar alıŖtırıldıđında iŖaretleli mi yoksa iŖaretsiz mi geleceđini gstermektedir.

rnek olarak yazılan bir metin kutusunun ve iŖaretleme kutucuđunun kodu Resim 9.1.1'de ve kodların alıŖtırılmıŖ halleri Resim 9.1.2'de yer almaktadır.

```
11 <form>
12
13 <input type="text" name="metinkutusu" value="Metni Buraya Girebilirsiniz">
14
15 <br>
16
17 <input type="checkbox" name="iŖaretleme kutusu" value="no">
18
19 <br>
20
21 </form>
```

Resim 9.1.1



Resim 9.1.2

Tıklanabilir düğmeler ve açılabilir menüler diğer form bileşenlerinin aksine açılan ve kapanan bir kod yapısına sahiptir. Her ikisi de diğer form bileşenleri gibi type, name ve value özelliklerini içerirler.

Tıklanabilir düğmeler **button** kodu altında yazılır. Type (tür) kısmında **submit** olarak belirtmeniz, butonun oluşması için gereklidir. Üzerine yazılmasını istediğiniz yazıyı value özelliğinin içerisine ekleyebilir ya da button kodunun kendi özelliğinden yararlanarak yazabilirsiniz.

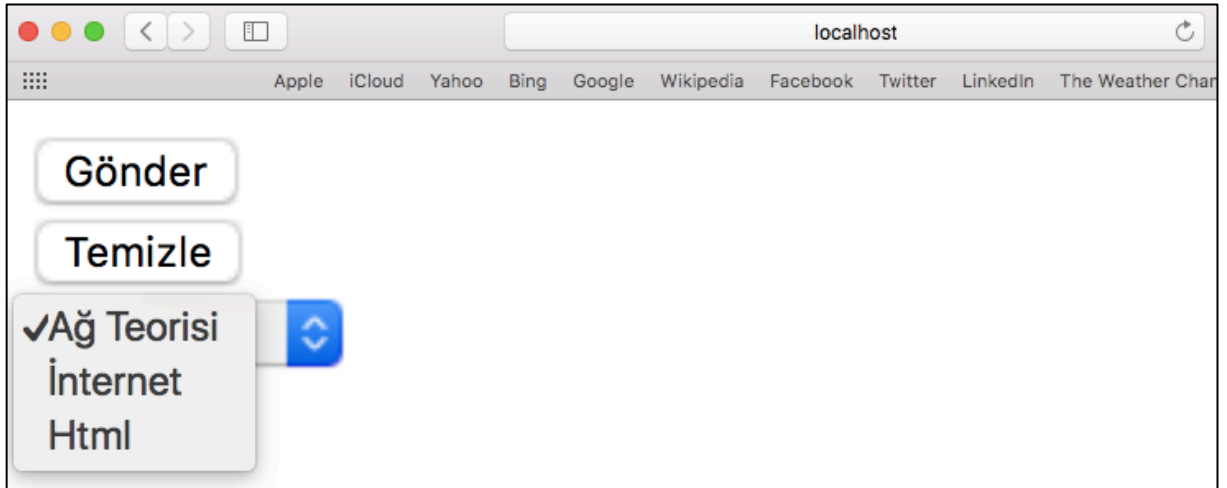
Button kodu içerisinde type kısmına **clear** olarak belirttiğiniz zaman metin kutusu içerisindeki yazıyı temizleyen bir türü bulunmaktadır.

Açılabilir menüler ise **select** kodu altında yazılır. Type (tür) kısmında **popup** olarak belirtmeniz, açılabilir menü olması için gereklidir. Ancak menülerin seçeneklerini oluşturabilmek için kendi içerisinde **option** kodunu ayrıca kullanmalıyız.

Gönder ve temizle düğmesi, ayrıca bir de açılabilir menü yaptığımız örnek kodlarımız Resim 9.1.3'te ve kodların çalıştırılmış halleri 9.1.4'te yer almaktadır. Eğer temizle düğmesini test etmek isterseniz, oluşturduğunuz metin kutusu ile birlikte çalıştırmanız gerekmektedir.

```
23 <form>
24
25 <button type="submit">Gönder</button>
26
27 <br>
28
29 <button type="clear">Temizle</button>
30
31 <br>
32
33 <select type="popup" name="ilkmenu" value="Lütfen Seçiniz">
34
35 <option value="ilk"> Ağ Teorisi</option>
36 <option value="ikinci"> İnternet</option>
37 <option value="üçüncü"> Html</option>
38
39 </select>
40
41 </form>
```

Resim 9.1.3



Resim 9.1.4

Web siteleri üzerinde gördüğümüz formlarda sadece form bileşenleri olmayıp, başında form bileşenlerinin niteliğini belirten yazılar da bulunmaktadır. Nitelikleri yazabilmeniz için html kodlarının temel özelliğinden yararlanarak, kodlarınızın başına ekleyebileceğiniz gibi, label kodundan yararlanarak da yapabilirsiniz.

Label kodunu kullanabilmek için form bileşenlerinin içerisinde belirttiğimiz ‘name’ özelliğinden yararlanmaktayız. Name kısmında belirttiğimiz ismi bildiğimiz zaman, label kodumuz o bileşene giderek, başına istediğiniz yazıyı eklemektedir. Örnek kodlarımız Resim 9.1.5’te ve kodlarımızın çalıştırılmış halleri Resim 9.1.6’da verilmiştir.

```
14 <form>
15 <!--Html kodları ile çalıştırılmış hali -->
16
17 Adres: <input type="text" name="metinkutusu" value="Metni Buraya Girebilirsiniz">
18
19 <br>
20
21 <button type="clear">Temizle</button>
22
23 <br>
24 <!-- Label kodu yardımı ile eklenen yazılar -->
25
26 <input type="checkbox" name="kutu1" value="no">
27 <label for="kutu1" > İstanbul</label>
28
29 <br>
30
31 <input type="checkbox" name="kutu2" value="no">
32 <label for="kutu2" > Ankara</label>
33
34 <br>
35
36 <select type="popup" name="menu" value="Lütfen Seçiniz">
37
38 <option value="ilk"> Ağ Teorisi</option>
39 <option value="ikinci"> İnternet</option>
40 <option value="üçüncü"> Html</option>
41
42 </select>
43
44 <br>
45
46 <button type="submit">Gönder</button>
47
48 </form>
```

Resim 9.1.5

The screenshot shows a web browser window with the address bar set to 'localhost'. The browser's menu bar includes 'Apple', 'iCloud', 'Yahoo', 'Bing', 'Google', 'Wikipedia', 'Facebook', 'Twitter', 'LinkedIn', and 'The Weather Cha'. The main content area displays a form with the following elements:

- A label 'Adres:' followed by a text input field containing 'Maltepe'.
- A 'Temizle' button.
- A checked checkbox next to 'İstanbul'.
- An unchecked checkbox next to 'Ankara'.
- A dropdown menu with 'Ağ Teorisi' selected, followed by the text 'Almak İsteddiğiniz Ders'.
- A 'Gönder' button.

Resim 9.1.6

Form oluşturduktan sonra, son olarak belirlenmesi gereken kısım ise formun oluşturulduktan sonra nereye yönlendirileceğidir. Bunun için form kodunun içerisine **action** kodu eklenmektedir ve karşısında gitmesini istediğiniz adresi **mailto** ile belirterek yazmanız gerekmektedir. Action özelliğinden sonra eklememiz gereken bir diğer özelliğimiz ise method özelliğidir. **method="post"** dersiniz, formu doldurduğunuzda ve gönder butonuna bastığımızda mail adresine yönlendirilecektir.

Formun oluşturulduktan sonra yönlendirilmesini sağlayan kodlarımız Resim 9.1.7’de yer almaktadır.

```
13  
14 <form action="mailto:sadievrenseker@gmail.com" method="post">  
15
```

Resim 9.1.7

10.BÖLÜM: E-POSTA (E-MAIL) NASIL ÇALIŞIR?

10.1 E-POSTA'NIN ÇALIŞMA DÖNGÜSÜ

10.2 SMTP (SIMPLE MAIL TRANSPORT
PROTOCOL)

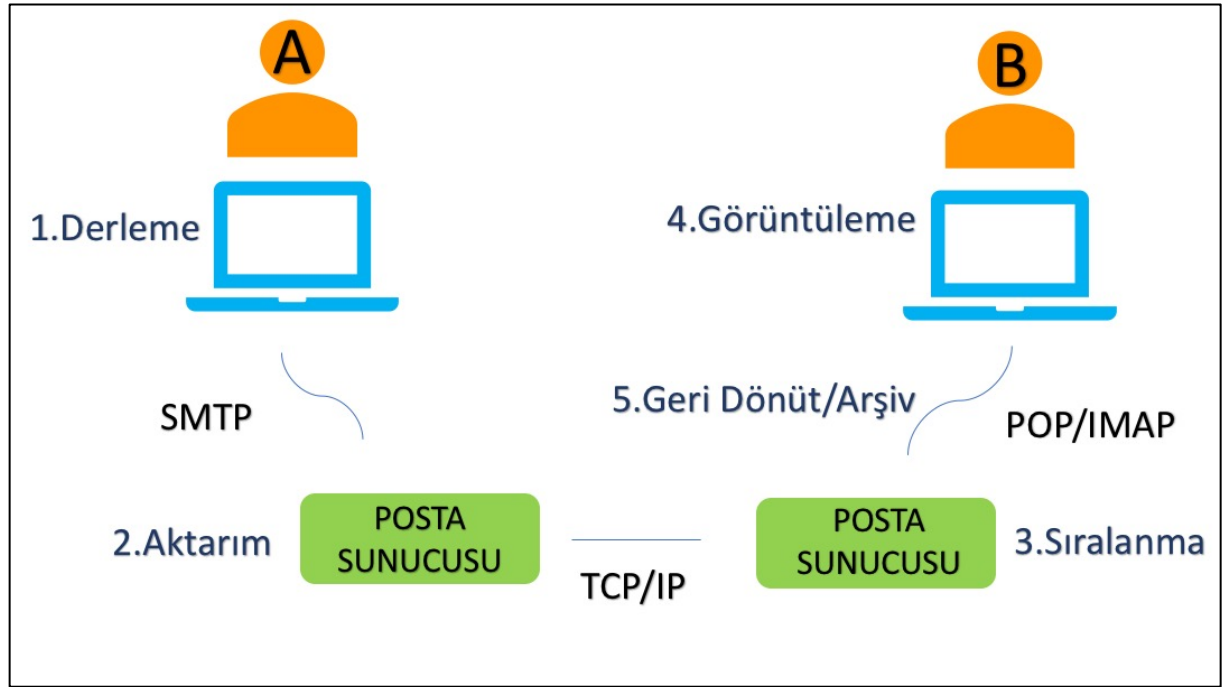
10.3 POP (POST OFFICE PROTOCOL) /IMAP
(INTERNET MAIL ACCESS PROTOCOL)

10.4 SPAM E-POSTALAR

10.1 E-POSTA 'NIN ÇALIŞMA DÖNGÜSÜ

E-Posta sistemi internetin ilk var olduğu zamanlardan bu yana çalışan bir sistemdir. Dolayısıyla günümüz dünyasında en çok kullanılan iletişim ağlarından da birini oluşturmaktadır. E-Posta sisteminin nasıl çalıştığını, bir mailin nasıl oluşturulduğunu, mailin içerisinde hangi bilgilerin yer aldığını açıklayacağız.

E-Posta sisteminin nasıl işlediğini anlayabilmek için öncelikle Resim 10.1.1'de yer alan bir postanın çalışma döngüsünü inceleyelim.



Resim 10.1.1

Görüldüğü üzere öncelikle, kullanıcı (A) kendi bilgisayarında bulunan posta istemcisinde (Outlook, Opera vb.) mail oluşturuyor ve oluşturulan mail bilgisayarda bulunan **SMTP** protokolü ile posta sunucusuna iletiliyor.

Eğer kullandığınız mail yazılımına web sürümü ile bağlantı sağlıyorsanız, ayrıca posta sunucusunun üzerinde web sunucusunun da bulunduğunu düşünebilirsiniz. Web sunucusu kullandığınız tarayıcı (Chrome, Safari, Yandex vb.) üzerinden sizin adınıza posta sunucusuna bağlanmaktadır.

Posta sunucumuz daha önce anlatmış olduğumuz **TCP/IP** protokolleri ile alıcının posta sunucusuna ulaştırılmaktadır.

B kullanıcısı, maillerini görüntüleyebilmek için **POP/IMAP** protokolleri ile posta sunucusuna bağlanarak, postalarını alabilmekte ve görüntüleyebilmektedir. Daha sonra kullandığı web ara yüzü aracılığıyla da yazılım ile geri dönüş sağlayabilmektedir.

10.2 SMTP (SIMPLE MAIL TRANSPORT PROTOCOL)

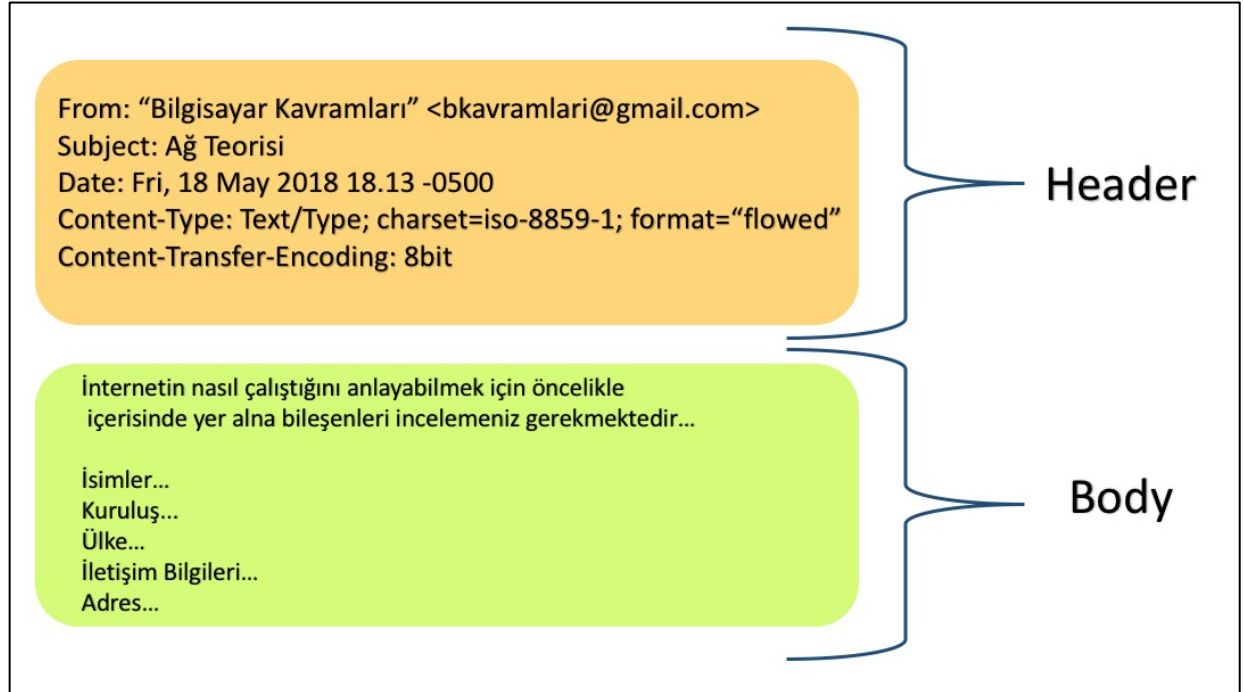
SMTP (SIMPLE MAIL TRANSPORT PROTOCOL), ASCII metin planlarına uygun olarak yazıldığı için, açıldığında bizim de okuyabileceğimiz anlaşılır metinler yer almaktadır.

Formatı zamanla değişebilmekle birlikte, içerisine eklemeler yapılabilecek ve Html formatına dönüştürülebilir bir yapıya sahiptir.

SMTP üzerinde herhangi bir güvenlik protokolü bulunmamaktadır. Ama bağlantılar üzerinde güvenlik sağlanabildiği için (TCP/IP), sizin sunucuya bağlanmanız güvenli hale getirilebilmektedir. Dolayısıyla güvenlik sağlayabilen portlarımız bulunmakla birlikte, sağlayamayanlar da bulunmaktadır.

10.2.1 SMTP FORMATI

SMTP yapısı incelendiği zaman, aynı Html kodlarında olduğu gibi bir **header** ve **body** kısmı bulunmaktadır. Header kısmında mesajın kime gönderildiği, konusu ve tarihi yer almaktadır. Body kısmı ise kullanıcı mesajı açtığı zaman göreceği bölümlerden oluşmaktadır. Herhangi bir metin editöründe açtığınız zaman, anlaşılabilir olarak görülen SMTP içeriği Resim 10.2.2’de yer almaktadır.



Resim 10.2.2

Ayrıca postalar SMTP ile oluşturularak gönderilse bile, alıcı tarafın posta sunucuları da SMTP formatının header kısmına eklemeler yapabilmektedirler. Bunlardan bir tanesi **Received** kısımlarıdır. Örneğin bir posta çeşitli nedenlerden dolayı, alıcı tarafın sunucusuna geç iletilir ve bu tarih kullanıcıya yansır. Olası aksaklıkların önlenmesi için alıcı tarafın posta sunucusu da üzerine tarih bilgisini eklemektedir. Received satırına tarih haricinde zaman çizelgesi (ne kadar sürede gönderildiği) ve kaynağı gibi bilgiler de eklenmektedir.

Ayrıca bir diğer ekleme yapabilen kısım ise Spam kontrolünü sağlayan yapılardır. Spam kontrollerini sağlayan yapılar hangi yazılım tarafından, hangi sürüm ile yapıldığını veya sonucu bize aktarmak için header kısmına bilgi ekleyebilmektedirler.

NOT: Spam herhangi bir mesajın, kullanıcının istek ve rızası olmadan aldığı postaların genel ismidir.

10.2.2 SAAT DİLİMLERİ (TIMESTAMP)

Postalarınızı gönderirken dünya üzerinde yaşanabilecek saat farkından dolayı SMTP aynı zamanda bu sorun içinde çözüm olmakta ve gerekli düzenlemeleri yapmaktadır.

Dünya üzerinde farklı ülkelerin saatlerini konumlarına göre ifade etmesi, İngiltere’de yer alan Greenwich noktasına göre düzenlenmektedir. Çoğu zaman gördüğümüz **GMT+3 (Greenwich Mean Time)** ifadesi o noktaya göre, 3. Saat dilimi içerisinde yer aldığınızı göstermektedir.

10.2.3 E-POSTA EKLERİ (E-MAIL ATTACHMENTS)

E-Postanızı göndermeden önce, postanıza ek olarak dosya (resim, metin vb.) gönderilebilmektedir. Fakat SMTP, daha önce de değinildiği üzere açık bir metin olarak düzenlenmektedir. Dolayısıyla resim dosyalarının da ek olarak gönderilebilmesini sağlamak için **MIME (Multipurpose Internet Mail Extensions)** geliştirilmiştir.

MIME, İnternet üzerinde kullanılan bir posta protokolüdür ve Türkçesi çok amaçlı internet posta uzantıları olarak çevrilmiştir. MIME, açık ve ASCII karakter kodları ile birlikte çalışan bir protokoldür ancak, ASCII karakterleri dışında da dönüşümler gerçekleştirerek (fotoğraf dosyaları) dosya gönderimi yapabilmektedir. SMTP dosyalarının body kısmına dosya eklemesi yapabilmektedir ve bu dosyalara resimler de dahil olabilmektedir.

NOT: Bu hususta ikilik tabana ait bilgileri, daha önce görmüş olduğumuz ASCII tablosu yardımı ile dönüştürülemez. Bu dönüştürme işlemi için özel olarak düzenlenen tablolar kullanılmaktadır.

Normal şartlarda resim dosyaları 1 ve 0'lerden oluşan dosyalardır. 1 ve 0'lerden oluşan ikilik sisteme ait bu bilgileri karakterlere çevirmekte ve karaktere çevrilebilen dosyayı da alıcının posta sunucusu alabilmektedir.

Örnek bir MIME mesajı Resim 10.2.3.1'de verilmiştir.

```
From: Sadi Evren SEKER <sadi@bilgisayarkavramlari.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="XXXXKESME"

Ornek bir mime postasinin basligi.

--XXXXKESME
Content-Type: text/plain

Postanin icerigi.

--XXXXKESME
Content-Type: text/plain;
Content-Disposition: attachment;
    filename="ilave.txt"

Posta ile gelecek olan ilave.txt dosyasinin icerigi.

--XXXXKESME--
```

Resim 10.2.3.1

Resim 10.2.3.1'de yer alan mesajı incelediğimizde, en üstte **MIME** mesajının sürümünün iletildiğini görebiliriz.

Hemen altında **Content – Type** kısmında mesajın içeriği hakkında bir bilgi yer almaktadır. Eğer bu kısımda text yazıyorsa metin, application olarak belirtildiyse bir uygulama, audio olarak yazılıyorsa ses dosyası, image ise bir resim olduğunu göstermektedir.

Resim 10.2.3.1'de text/plain olarak belirtilmesinin sebebi, ilk kısmın dosyanın ne olduğunu, ikincisi ise türünü (alt başlığını) göstermektedir.

Fakat bazı dosyaların gönderilmesi engellidir. Örneğin oluşturulan exe. dosyalarının çoğu kötü amaçlı yazılımlar için kullanıldığından ve bu dosyaların açılmadan taranamaması dolayısıyla birçok program tarafından gönderilmesi engellenmiş bulunmaktadır.

Boundary karşılığında bulunan ifade, mesajın birbirinden ayrıldığı noktaları göstermektedir.

10.3 POP (POST OFFICE PROTOCOL) /IMAP (INTERNET MAIL ACCESS PROTOCOL)

POP (Post Office Protocol) ve **IMAP** alıcı tarafında çalışan protokollerdir. POP, bir post ofisinin olmasını sağlamaktadır. Çalışma mantığını bir örnek ile açıklamaya çalışalım. Örneğin internet bağlantınız normale göre daha yavaş ise, hızlı ağa bağlanamıyorsanız ya da ara ara bağlanabilen bir yapıya sahipseniz, POP gereği mailleriniz, internet bağlantısının sağlandığı an itibari ile postaları, posta ofisine gönderiyor. Postalar, posta ofisinde depolanıyor ve daha sonra bağlantınız düştüğünde okuyabilmenizi sağlamaktadır.

IMAP (Internet Mail Access Protocol), POP'dan farklı olarak postaları sunucu tarafında bekletiyor ve posta sunucusuna bağlandığınız zaman okuyabilmektesiniz. Dolayısıyla yaptığımız her hareket (postaya cevap verme, yönlendirme, dosya ekleme) bir internet trafiğine sebep olmaktadır.

POP üzerinde işlem yaparken ise bilgisayar kendi içerisinde (yerelde çalıştığı için) işlemleri düzenleyebilmektedir. Yalnızca verileri alırken ve gönderirken internete ihtiyaç duyulmaktadır.

10.4 SPAM E-POSTALAR

Spam e-postaları ilk gelişmeye başladığı zamanlarda telefonlar üzerinden yapılmaktaydı. Daha sonra spam e-postalarının bir tehlike oluşturduğu ortaya çıkmış ve mail şirketleri bunlardan sorumlu tutulduğu için çözüm bulunması konusunda çalışmalara başlanmıştır.

Günümüzde spam e-postaları büyük bir sorunu teşkil edilmektedirler ve spam sorununun önüne geçilebilmesi için yüksek maliyetler ile ödemeler yapılmaktadır.

Spam e-postalarını engelleyebilmek için bulunan çözümler arasında ilki, spam adresi olduğu düşünülen e-posta adresini engellemektir. Bu çözüm günümüzde hala birçok kullanıcı tarafından kullanılmaktadır.

Diğer bir yöntem ise kelimeler üzerinden yapılan engellemelerdir. Belirlenen kelimelerin yer aldığı e-postalar ve e-posta adresleri otomatik olarak engellenmesi sağlanmaktadır.

Bu tip yöntemler günümüzde hala kullanılmakta olup, büyük firmalar tarafından daha gelişmiş çözümlerde bulunmuştur. Örneğin yüksek teknolojiye sahip olan firmalar, belirli bir adres üzerinden yüksek sayıda mail adresine gönderim yapıyorsa, o adres spam adresleri listesine alınmaktadır.

Spam postalarının sistemine baktığımız zaman, alıcı tarafı korumak amaçken, aynı zamanda göndericiyi de korumak amaçlanmaktadır. Örneğin posta adresinize bulaşmış olan bir zararlı yazılım dolayısıyla, sizin posta adresiniz üzerinden de spam postaları atılabilmektedir. Bu gibi durumların önlenmesi için gönderici adreslerinin de korunması gerekmektedir. Bu gibi durumlarda göndericinin hesabı bloke edilerek, kullanıcının adresinde zararlı yazılım olduğu fark etmesi sağlanır ve önlem alması için çalışmalarda bulunulur.

11.BÖLÜM: İNTERNET GÜVENLİĞİ VE KİŞİSEL ÖNLEMLER

11.1 CHAINMAIL

11.2 URBAN LEGEND

11.3 PHISHING

11.4 PHARMING

11.5 SERTİFİKALAR

11.6 KÖTÜ AMAÇLI YAZILIMLAR (MALWARE)

Bu bölümde internet üzerinde kişisel bilgisayar kullanıcısı olarak kendi güvenliğini nasıl sağlarsınız sorusuna cevap verilecektir.

11.1 CHAINMAIL

Chainmail, internet üzerinde postalarda, mesajlarda, gönderilerde yer alan, size vaatlerde bulunan ya da bir şekilde ilginizi çekmeye çalışarak, sizi gönderiyi bakmaya yönelten posta zincirleridir.

Bu tip postalar iyi amaçlı olabileceği gibi çoğunluğu kötü amaçlı ve güvenliğinizi tehdit edecek şekilde yapılandırılmışlardır. Örneğin sosyal ağlarda rastladığınız ve tıkladığınız bir gönderiden yola çıkarak para kazanabilecekleri gibi, sizin üzerinizden analiz çıkarmaları da mümkündür. Bu gibi durumlar daha sonradan size tehdit oluşturabilecek formlara dönüştürülebilirler.

Örneğin size gelen mesajlarda ‘bu mesajı 10 kişiye gönderin, çekilişe katılın’ vb. ifadelerde, mesajı ilettiğiniz 10 kişi belirlenerek, daha sonra onların kişisel bilgilerine erişebilmektedirler. Kişisel verilerine eriştikleri arkadaş çevreniz tarafından (onlarmış gibi) size mesaj atabilmeleri ve yönlendirmeleri mümkündür.

Chainmail’ler kötü amaçlı olarak kullanılmadığı durumlarda bile, sizin internetinizi, kaynaklarınızı ve vaktinizi harcaması söz konusudur.

11.2 URBAN LEGEND

Urban Legend kavramı internet üzerinde sansasyonel haberler yaratan ve gerçeği yansıtmayan haberler içeren kaynaklara verilen isimdir.

Bu gibi kaynakların amacı, espri amaçlı olabileceği gibi çoğu zaman yalan haberleri yaymak için de kullanılmaktadırlar. Urban Legend kaynaklarının başlangıcı ve en yaygını olarak bilinen snopes.com’dur. Ayrıca Türkiye’de bulunan zaytung.com’da örnek olarak gösterilebilmektedir.

Bu gibi kurulan kaynakların amacı sosyal mühendislik için kullanılmaktadır. Sosyal mühendislikler kişisel ve toplumsal saldırılar olmak üzere iki türe ayrılmaktadır. Toplumsal olarak düzenlenen saldırılarda, inanların topyekun aynı yanlış fikre yönlendirilmesi ve yanıltılması hedeflenmektedir.

11.3 PHISHING

Türkçede kandırılmak anlamına gelen ve İngilizceden çevrildiğinde ‘oltaya gelmek’ olarak karşılanan **Phishing** kavramı, size bir bilgi gelmesi ile aksiyon almaktadır. Örneğin telefonunuza gelen ‘kredi kartınız 3 gün içerisinde iptal edilecektir, yardımcı olabilmemiz için aşağıdaki formu doldurun’ gibi bir mesaj buna örnek olarak verilebilmektedir. Kullanıcının o anda panik haliyle ya da dikkatsizliği ile kişisel bilgilerinin alınması ve gönderilmesi temeline dayanmaktadır.

Bunun başka bir yöntemi ise siteler üzerinde kullanılan üyelik işlemleridir. Girdiğiniz bir siteye üye olduğunuz zaman, form doldurulmanız istenmekte ve bazı kişisel bilgilerinizi almaktadırlar. Bu sitenin kendi tarafından tehdit amaçlı kullanılabilmesi gibi, site içerisine sızan başka kişilerden de kaynaklanabilmektedir.

Bu hususta alabileceğiniz en önemli önlem, güvenilir olmayan sitelere kişisel bilgilerinizi girmemeniz ve aynı zamanda mümkün olduğunca az siteye bilgilerinizi vermenizdir.

Çocuk, genç, yetişkin ya da yaşlı demeden herkesin oynadığı oyunlar bu gibi tehdit oluşturabilecek yazılımlar için bir saha oluşturmaktadır. Örneğin bazı oyunlar ücretsiz kategorisinde yer aldıkları halde, sevilerin atlanması ile birlikte ücretli hale gelebilmektedir ya da oyunun belirli bölümlerine, araçlarından para alınabilmektedir.

Oyunlarda dahil olmak üzere bilgisayarınıza kurduğunuz her uygulama, sizin için bir tehdit oluşturabilir. Bilgisayarınız üzerinden kişisel bilgilerinizi toplaması mümkündür. Örneğin, çok popüler bir uygulama olan Angry Birds oyununu yazan firma, yaptığı açıklamada, oyunun içerisinde back-door bırakıldığı ifadesine yer vermiştir. Türkçeye açık(arka) kapı olarak çevrilebilen bu terim, bilgisayarınız içerisinden kişisel bilgilerinizi sızdırıldığını ifade etmektedir.

11.4 PHISHING İÇİN ALINABİLECEK ÖNLEMLER

İnternet üzerinde kişisel bilgisayar kullanıcısı olarak güvenliğinizi sağlayabilmeniz için yapmanız gereken temel hareketler aşağıda sıralanmıştır.

- Sizden bilgi ya da herhangi bir şey talep eden gönderilere karşı her zaman durun ve düşünün. Örneğin size bir mail geldi, bu maili size gönderen ile bağlantınız var mı? Kişisel bilgilerinizi daha önce paylaşmış mıydınız? Sizinle neden mail ile iletişim kurmayı tercih ettiler? Vb. soruları cevaplayarak karar vermeniz gerekmektedir.
- Daha sonradan kişisel bilgilerinizin satılmasına ya da kullanılmasına karşılık olarak, bu bilgilere erişmek isteyen kuruluşlara ait internet ve gizlilik politikasını mutlaka gözden geçirin.

11.5 PHARMING

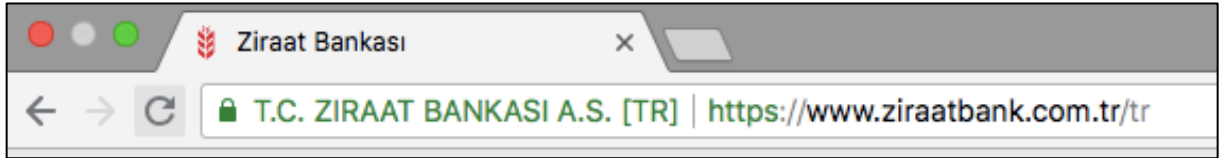
Pharming kavramı, nternet korsanları tarafından, bilinen bir web sayfasını gerçeğe en yakın şekilde kopyasını oluşturarak kullanıcıya sunması anlamına gelmektedir.

Web adresini de orjinaline yakın olarak aldıkları için bunu kolaylıkla fark edememektesiniz. Daha sonra sitenin orjinalinde yer aldığı gibi, çıkan giriş ekranına bilgilerinizi girdiğiniz zaman kişisel bilgilerinize erişebilmektedirler.

Kişisel bilgilerinizi girdiğiniz zaman yazan ‘siteye şu anda ulaşlamıyor’ vb. gibi bir mesajı size gösterebilirler. Bu zaman zarfında orijinal site üzerinden sizin kişisel bilgileriniz yardımı ile giriş yapmış olacaktır.

11.6 SERTİFİKALAR

Kişisel bilgilerinizi girmenizin gerekli olduğu sitelerde güvenliğinizi sağlandığından emin olmak için, tarayıcınızın adres kısmında kilit işareti bulunmaktadır. Kilit işaretinin kapalı olması (Resim 11.6.1), verilerin direk olarak gönderildiği ve 3. Bir şahısın, siteyi dinleyebilse bile verilerinize ulaşamayacağı anlamına gelmektedir. Yani site içerisinde Http ya da Https gibi güvenlik önlemlerini içeren sertifikaların sağlandığı doğrulanmıştır.



Resim 11.6.1

Aksi durumda yani kilit simgesini açık olduğu durumlarda (Resim 11.6.2) ise, siteyi dinleyen 3. Bir şahıs tarafından verilerinize erişebileceği anlamına gelmektedir.

11.7 KÖTÜ AMAÇLI YAZILIMLAR (MALWARE)

Malware (**Malevolent Software**), bilgisayarınıza kurduğunuz uygulamalar içerisinde kötü amaçlı yazılım barındıranları kapsamaktadır.

11.7.1 KÖTÜ AMAÇLI YAZILIM TÜRLERİ

Virüs: Genellikle bilgisayar yazılımına zarar vermek amacı ile yapılan ve bilgisayarlara bulaşabilen zarar verici bir yazılım türüdür. Bilgisayarınızda yer alan uygulamaların olağandışı davranmasına sebep olurlar.

Kurduğunuz bir yazılım üzerinden ya da usb vb. araçlar yardımı ile de gelebilmektedir. Her bir ortama kendilerini kopyalayarak, o ortama kendilerini yayabilmek için yollar geliştirmektedirler. Örneğin, 'bilgisayarınızda virüs bulundu' şeklinde açılan bir pencere yardımı ile indirdiğiniz antivirüs programlarının kendileri virüs içerebilirler.

Solucan (Worm): Bilgisayarda bulunan ve kendilerini yayan zararlı yazılım çeşididir. Virüs gibi davranırlar. Amaçları bilgisayarınıza zararlı yazılım bulaştırmaktır.

Virüs ve solucanların kendi aralarında çeşitleri bulunmaktadır ve hepsi farklı bir amaca sahiptirler. Örneğin bazıları dosyanızı bozarken, bazıları ise dosyalarınıza eklemeler yapmaktadırlar.



Resim 11.7.1.1

Truva Atı (Trojan): Türkçesi Truva Atı olarak da geçen Trojan, güvenli gibi gözükse fakat yazılım açıldığı zaman bilgisayarınızın içerisine kötü ve zararlı yazılımlar bulaştıran yapılardır.

Adware (Reklam Yazılımı): Bilgisayarınızın açılış ekranında ya da web sayfaları üzerinde sürekli olarak reklam panolarının açılıyor olması, bilgisayarınıza adware yazılımını girmiş olmasından kaynaklanmaktadır.

Spyware (Casus Yazılımı): Bilgisayarınıza yayıldığı zaman, kişisel bilgilerinize ulaşmak ve bunları toplamak için bilgisayarınıza kurulmuş yazılımlardır. Örneğin bilgisayarınızın kamerasını kullanabilir ya da girdiğiniz bilgileri okuyabilir.

Rootkit: Bilgisayarın yönetici yetkisini ele geçiren yazılımlardır. Bunlar sizin bilgisayarınız üzerinden saldırı yapmak vb. için kullanılabilir.

NOT: Kötü amaçlı yazılımlar sürekli olarak geliştirilmektedirler. Sürekli olarak arkadaşlarınız ya da kuruluşlar tarafından geliyormuş gibi gösterilen postalar vb. bunlara örnek olarak gösterilebilmektedirler.

Bunların yanında çok üst düzey kötü amaçlı yazılımlar da bulunmaktadır. Örneğin Zero Day Attack (Sıfır Gün Açığı) adı verilen bir saldırı çeşidi, bilgisayarınızın güncellenmesi sırasında, işletim sisteminizin bir açığından faydalanılarak zararlı yazılımları bilgisayarınıza bulaştırabilmektedir.

Bu gibi kötü amaçlı yazılımlardan korunmak için öncelikle yüklemiş olduğunuz her yazılımı detaylıca araştırmanız gerekmektedir.

Ayrıca Usb üzerinden zararlı yazılımları bulaştırmak, en yaygın yöntemler arasında yer alırken, hiç kimseye ait bir aygıtı taratmadan bilgisayarınıza bağlanmasına izin vermeyin.

Bilgisayarınız üzerinden sürekli yedeklerini almanız da kötü amaçlı yazılımların etkilerinden an az zararlı korunmanızı sağlamaktadır.

11.7.2 KÖTÜ AMAÇLI YAZILIMLAR NASIL YAYILIR?

1. Postalarınız üzerinden yayılım sağlayabilirler. Örneğin size gelmiş bir postada ek olarak gönderilen bir zip dosyasını açtığınız zaman bilgisayarınız içerisine yayılım sağlayabilmektedirler.
2. Bir Usb aygıtı üzerinde barınan Malware, bağlandığı tüm cihazlar üzerinde yayılım yapabilmektedirler.
3. Uygulamalarınız ya da işletim sistemleriniz üzerinde yer alan açıklardan faydalanarak, yapılacak olan bir sonraki güncelleme ile kötü amaçlı yazılımları bilgisayarınıza bulaştırmaları mümkündür.
4. İnternet üzerinden indirilen her türlü uygulama, güvenilir olduğundan emin olunmadığı sürece, indirilmemelidir. Birçoğunun içerisinde yer alan kötü amaçlı yazılımlar, ilk zamanlarda kendilerini belli etmeseler de, daha sonralarda bilgisayarınız üzerinde olağandışı hareketlere sebep olabilmektedirler.

11.7.3 KÖTÜ AMAÇLI YAZILIMLARA KARŞI ALINABİLECEK ÖNLEMLER

1. Güvenmediğiniz ve araştırmadığınız hiçbir yazılımı bilgisayarınıza indirmemelisiniz.
2. Bilgisayarınızda bulunan uygulamaları güncel tutmalısınız. (Yazılımların kendi içerisinde açıkları olabilmektedir. Firmalar daha sonradan bunları fark ederek, yazılımlarını güncelleştirmekte ve giderebilmektedirler.)
3. Bilgisayarınız üzerinde bir güvenlik duvarı (Firewall) kullanabilirsiniz. Kuracağınız güvenlik duvarı üzerinden bilgisayarınızda yer alan uygulamaların o anki hareketlerini izlemeniz mümkündür. Ne kadar paket aldı, gönderdi vb. bilgileri incelerseniz, olağandışı durumlardan şüphelenerek ve araştırarak önlemler alabilirsiniz.
4. Şifrelerinizi rastgele üretebilir ya da tamamıyla size ait olarak geliştirilen bir şifre edinerek, her yerde farklı şifreler girerek korunabilirsiniz.
5. Bilgisayarınız üzerinde ihtiyacınız olmayan uygulamaları kapatın ve hatta kaldırın. Uygulamalar kullanılmasa bile bilgisayarınız üzerinde kurulu olduğu için saldırı ihtimaline karşılık olarak tehlike yaratabilmektedir.
6. Kötü amaçlı yazılımlardan korunabilmek veya etkisini en aza indirebilmek için bilgisayarınızın ve diğer tüm dijital aygıtlarınızın yedeklerini almalısınız.

12.BÖLÜM: İNTERNET ÜZERİNDE ARAMA NASIL YAPILIR?

12.1 VERİ KOLEKSİYONU (DATA COLLECTION)

12.2 ARAMA MODELİ

12.3 OTARİTE BAZLI ARAMA (AUTHORITY
BASED SEARCH)

12.4 ETKİLİ ARAMA TEKNİKLERİ

12.1 VERİ KOLEKSİYONU (DATA COLLECTION)

Web Crawler, Bot ya da **Spider** olarak bilinen kavramalar birer web robotlarıdır. Ve bu robotlar internet üzerinde birbirine bağlı olan ağda bir sayfadan başlayarak linkleri takip ederek devam etmektedir.

Dokümanların birbirine bağlı olduğu bu tip ağlarda diğer kullanıcılar da yer almaktadır. Daha iyi açıklayabilmek için, örneğin ağlar üzerinde link takibini yapan ve sayfaları dolaşan kişinin bir robot yazılım olduğunu düşünün. Web sitelerinin içeriklerini alarak kendisine bir indeks oluşturmakta ve bundan yararlanarak ilerleyen zamanlarda kullanıcıların önüne bu sayfaları sunabilmektedir.

Bu bölümde birkaç tane sorun ortaya çıkmaktadır. Bunlardan ilki internet ile ilgili **gerçeklik (veracity)** problemi. Gerçeklik probleminde, bir siteden diğerine bilginin değişebilirliği söz konusu olabildiği gibi (sitelerin ne kadar zaman aralığında güncellendiğine bağlı olabilir), girdiğiniz bir siteye, tekrar ulaşamamanız (sitenin gerçekliği) da söz konusu olabilmektedir. Ayrıca girdiğiniz bir sayfaya tekrar erişememenizde mümkündür. Bu tip sayfaları hangi periyotlarda takip etmeniz gerektiğini ve sayfa üzerinde sadece değişikliği fark etmesini sağlayan web crawler adında yazılımlar bulunmaktadır.

Web crawler topladıkları bilgiler ile bir indeks oluşturduğu ve bu bilgileri diskte değil de Ram üzerinde tuttuğu için istediğiniz anda hızlı bir şekilde erişim elde etmenizi sağlayabilmektedirler.

NOT: Sunucu Çiftlikleri (Server Farms), birçok sunucunun bir arada bulunduğu yerlerdir. Sunucular Ram üzerinde bilgileri tutarak, anlık olarak gelen bir isteğe cevap vermek için hazırda beklemektedirler.

12.2 ARAMA MODELİ

Arama motoru size belirli aramalarınız sonucu çıkacak sonuçları hazırlamaktadır. Bir de işin diğer kısmı, arama çubuğuna yazdığımız anahtar kelimeyi nasıl getirileceğine göre değişik şekiller alabilmesidir. Hangi site en üstte yer almalı, hangisi kullanıcının ilgisini daha fazla çekebilir, hangi site aranan kelimeler ile daha ilgilidir vb. gibi sonuçlar yer almaktadır.

NOT: Tarayıcılar bu sorulara göre geliştirdikleri algoritmaları paylaşmamaktadırlar. Her tarayıcının kendine özgü olarak geliştirilmiş algoritmaları bulunmaktadır. Aynı zamanda bu sonuç gösterme yöntemlerini kişiye özgü olarak da özelleştirebilmeleri mümkündür

12.2.1 KELİME HAFIZASI

Kelime kutuları olarak düşünebileceğiniz bu yapı Google olmadan önce ve sonra olmak üzere iki parçaya ayrılabilir.

Genel yapısı itibari ile kelimelerin kategoriler halinde parçalara ayrılıyor olmasından oluşmaktadır. Örneğin bir spor sitesinde geçen kelimeler, **geçme sıklığı (Frekans Özelliği – Frequency)** baz alınarak bir kutuya girebilmekte veya magazin sitesinde geçen kelimeler başka bir kutuda yer alabilmektedir.

Arama çubuğuna girdiğiniz anahtar kelimelerden yola çıkarak, sitenin hangi konu ile ilgili olduğu ayırt edilebilmektedir.

Bu kısımda köklerine (stem) ayrılmasını sağlayan bir kelime dedektörünü kullanabilmek mümkündür. Bu yöntem ile kelimelerde yer alan yapım ve çekim ekleri algılanabilmektedirler. Çekim eklerini kelimedenden çıkartarak kelimenin köküne inilmektedir. Böylece kelimenin anlamı öğrenilerek, doğru kategoriye ulaşması sağlanmaktadır.

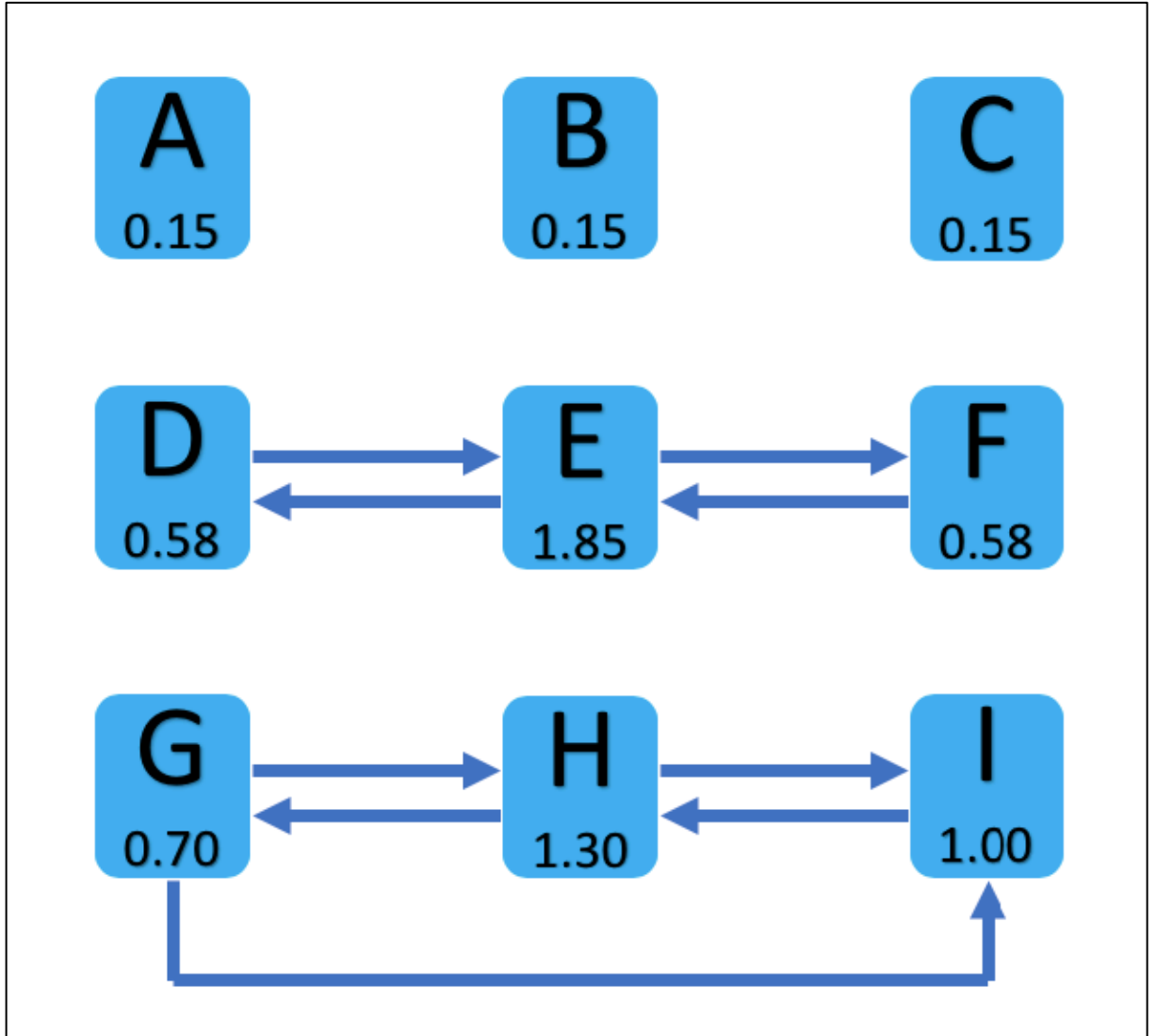
NOT: Yapım ekleri, kelimelere eklendiği zaman kelimenin anlamını değiştiren eklerdir.

Çekim ekleri ise kelimelere eklendiği zaman anlamı değiştirmeyen eklerdir. Dolayısı ile çekim eklerinin çıkarılması, kelimenin köküne inebilmek için yardımcı olmaktadır.

12.3 OTORİTE BAZLI ARAMA (AUTHORITY BASED SEARCH)

İnternet üzerinde bir otorite oluşturulmakta ve sayfalar aslında bir otorite belirtmektedirler.

Sayfa (Page) kelimesi, Google arama motorunu kuran o zamanlar üniversite çağında olan iki gencin bir projesi olarak ortaya çıkmıştır. Projede yeni olan ve daha önce geliştirilmeyen kısım **Page Rang** algoritmasıdır. Bu algoritmayı kısaca açıklamak gerekirse, site ne kadar fazla link alırsa (tıklanırsa), sitenin değeri o kadar artırılmaktadır.

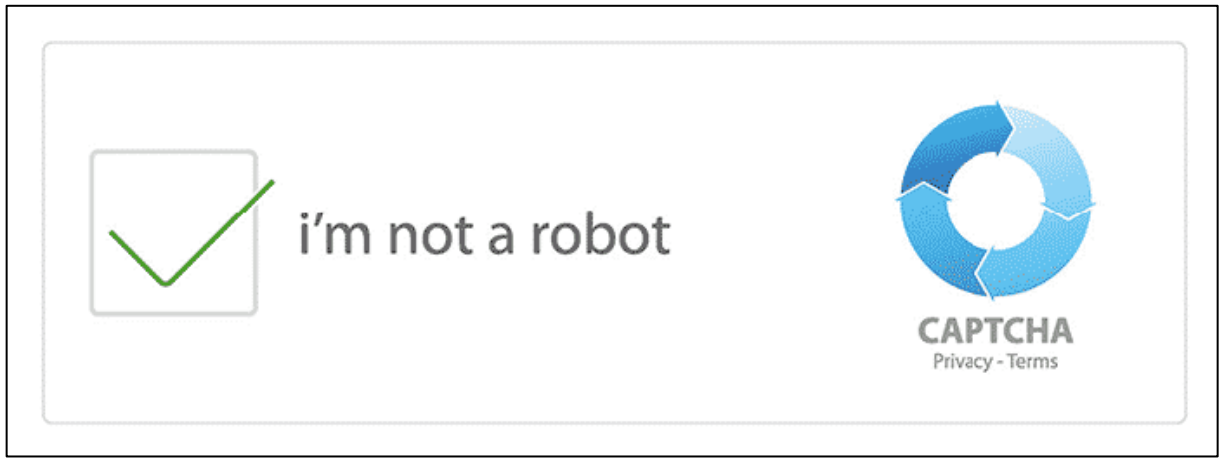


Resim 12.3.1

Resim 12.3.1’de verilen örneğe göre 9 tane web sayfası arasında en fazla değere sahip olanların, en fazla link aldığını da görebilirsiniz.

Page Rang algoritması ile birlikte bazı sorunlarda ortaya çıkmaktadır. Bunlardan en önemlisi spam sayfalarının oluşmaya başlamasıdır. Spam sayfaları da page rang algoritmasından yüksek değer alarak ve hatta yüksek değer almak için para ödeyerek sıralamada üst kısımlarda yer alabilmektedirler. Spam sayfalarının sorununu ortadan kaldırılabilmek için **CAPTCHA** denilen bir sistem oluşturulmuştur.

CAPTCHA açılımı '**Completely Automated Public Turing Test To Tell Humans And Computers Apart**' olan ve Türkçeye 'İnsan ve bilgisayarları ayırt edebilen, tamamen otomatize edilmiş açık Turing testi' olarak çevrilebilmektedir. Turing testleri Resim 12.3.2'de örnekleri yer alan, insan ya da makineleri ayırt etmeye yarayan testlerdir.



Resim 12.3.2

Robotların oluşması ve çoğalması, siteler üzerinden bilgi toplamalarına karşılık olarak, linklerin takip edilmesini engellemek için, siteler belirli protokollere uymaktadırlar. Diğer bir yöntemi ise kodlarınıza ‘**nofollow**’ özelliğini tanımlayabilirsiniz (Resim 12.3.4)

```
<body>  
  <a href="www.sadievrenseker.com" nofollow="nofollow" >  
</body>
```

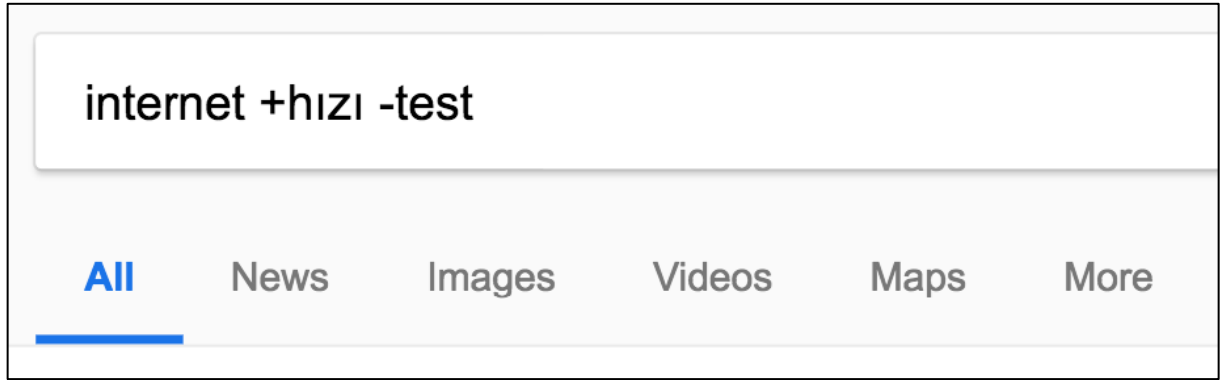
Resim 12.3.3

NOT: Sitelerin içerisinde erişilmesini (bilgilerin kopyalanmasını) istemedikleri yerleri görebilmek için, web sayfası adresinin ardından **/robots.txt** yazmanız yeterlidir. Örneğin Beyaz Saray’ın sitesi için, bilgilerin alınması engellenen yerleri www.whitehouse.gov/robots.txt yazarak görebilirsiniz.

12.4 ETKİLİ ARAMA TEKNİKLERİ

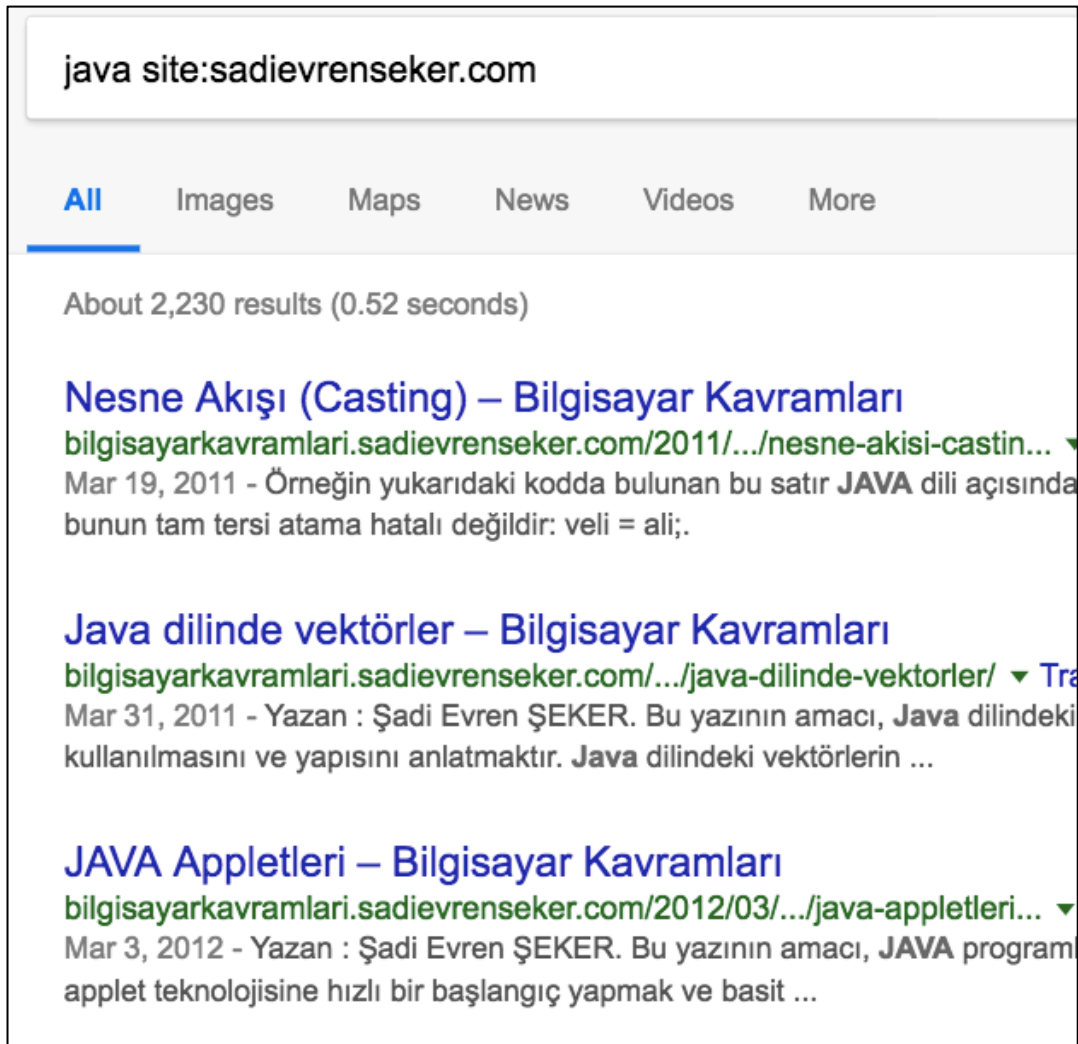
Bu bölümde tarayıcınız üzerinde arama yaparken sonuçlarınıza daha net bir şekilde istediğiniz gibi ulaşabilmeniz için sağlanan bazı kolaylıklardan bahsedilecektir.

- Anahtar kelimeleriniz arasından, sitenin ilgili olmasını istediğiniz ifadenin başına “+” (artı), istemediklerinize ise “-” (eksi) işareti koyarak ayrıştırabilirsiniz mümkündür (Resim 12.4.1).



Resim 12.4.1

- Eğer aramanızı sadece bir site içerisinde yapmak istiyorsanız başına **site:** yazarak girmeniz, tarayıcınızın o sitede arama yapmasını sağlayacaktır (Resim 12.4.2).



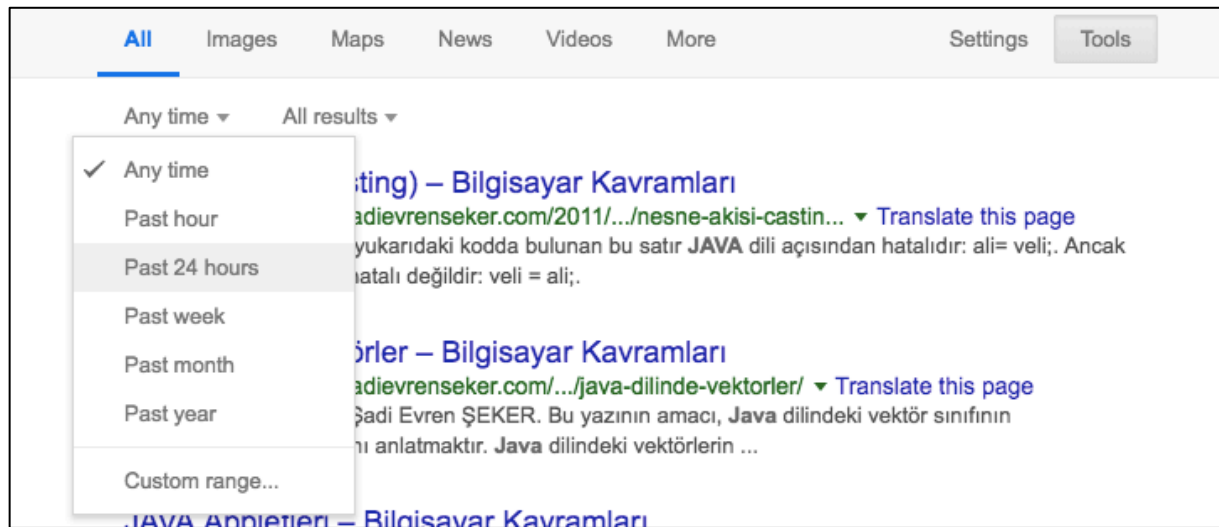
Resim 12.4.2

- Tarayıcınız üzerinde yer alan (örnek olarak Google Chrome gösterilmiştir – Resim 12.4.3) ‘daha fazla (more)’ bölümünden istediğiniz arama içeriklerini seçerek, siteler arasından eleme yapılmasını sağlayabilirsiniz.



Resim 12.4.3

- Tarayıcınızda bulunan ‘search tool’ bölümünden aramanızı, en son kaç saat içerisinde ya da kaç hafta içerisinde yapılmasını istediğinizi belirleyebilirsiniz (Resim 12.4.4).



Resim 12.4.4

13.BÖLÜM: ŞİFRELEME/KRİPTOLOJİ

13.1 KRİPTOLOJİ BİLİMİ

13.2 SEZAR ŞİFRELEMESİ

13.3 ONE TIME PAD

13.4 ETKİLİ ARAMA TEKNİKLERİ

13.5 ANAHTARLARIN DAĞITILMASI

13.1 KRİPTOLOJİ BİLİMİ (CRYPTOLOGY)

Kriptoloji kelimesinin Türkçe karşılığı şifre bilimidir. Anlatılmak istenen ifadenin belirli bir sisteme göre şifrelenmesi, güvenli bir şekilde karşı tarafa iletilmesi ve karşı tarafın bu mesajı deşifre edebilmesi anlamını taşımaktadır.

Günümüzde teknolojinin hızla gelişmesi üzerine internet üzerinde oluşan güvenlik açıkları, kişisel hayatımızda olduğu kadar, özellikle devlet kurumlarında ciddi sorunlar teşkil etmektedir.

Geçmiş zamanlardan günümüze kadar geldiğimizi zaman ilk olarak karşımıza Sezar şifrelemesi çıkmaktadır.

13.2 SEZAR ŞİFRELEMESİ

Tarihin ilk şifrelemesi olarak da geçen **Sezar şifrelemesi**, kaydırmalı şifrelemenin (Shift Chiper) bir türüdür.

Basitçe anlatılmak gerekirse, anlatılacak olan ifadede yer alan karakterlerin, belirlenen anahtar kadar kaydırılması ile ifade edilmektedir. Deşifre edilirken de aynı şekilde, anahtar kadar karakterler geri sarılmaktadır.

Gönderilecek Mesaj: Bilgisayar Kavramaları

Anahtar (Key): 2 (Her harf, iki sonraki harfe kaydırılması belirlenmiştir)

Şifreli mesaj: Dknhkctcaş Mezşcoencşj

Genelde bu tip kaydırılma şifrelemeleri, kelimeler arasındaki boşlukların veya kolay kelime boyutlarının anlaşılması sağlanan yöntemler geliştirilmiştir. Bu yöntemlerden en çok kullanılanı frekans saldırısıdır.

Bir alfabede en fazla ve en az kullanılan harf sayısı bellidir. Bundan yola çıkarak **frekans saldırısına** göre, iki iletişim noktası arasında iletilen mesajlar toplandığı ve analiz edildiği zaman, en fazla kullanılan harfi, yüksek ihtimalle o dilde en fazla kullanılan harfe karşılık gelmektedir.

Bundan yola çıkarak, örneğin metinde şifreli metinde 'h' harfi çok fazla kullanılıyorsa (Türkçede en çok kullanılan harfin e olduğunu varsayalım) e harfine karşılık geldiği düşünülmektedir. Yapılan çıkarımlar sonucunda anahtar bulunmakta ve şifrenin geri kalanı çözümlenmektedir.

13.3 ONE TIME PAD

One Time Pad, iletilecek olan bir mesajın, şifrelenmiş bir mesaj ile birleştirilmesi ile oluşturulan bir şifreleme yöntemidir. Örneğin;

Gönderilecek Mesaj: This is another secret message.

Belirlenen Şifre : Vxdx ch yzez ymu kxuyra.

+

Şifreli Mesaj : Oelp kz ymssfql cbwpvt kvllkxi.

Her harfin alfabe de kaçınıcı harfe karşılık geldiđi bulunarak toplanır ve şifreli mesaj oluşturulur.

One Time Pad eđer bir kere gönderilen bir işlem ise şifrenin çözülememesi sorunu ortaya çıkmaktadır. Şifrenin çözülebilmesi için karşılıklı geri dönütlerin birden fazla kez gerçekleşmesi gerekmektedir.

One Time Pad yönteminin kullanılabilmesi için bir şifreye ihtiyaç duyulmaktadır ve şifreyi her iki tarafında bilmesi gerekmektedir. Her yeni mesajda yeni bir şifre oluşturulması gerektiđi için bu da ayrıca bir sorun teşkil etmektedir.

13.4 ANAHTARLARIN DAĞITILMASI

İletişim kurmak için oluşturduğumuz mesajların, maillerin ya da web sitelerine bağlanmak için gerekli izinlerin, şifreleme ile bilgisayarlara ulaştırılabilmesi konusunda günümüzde yaygın olarak kullanılan iki yöntem bulunmaktadır.

İlk yöntem her iki tarafta da üreteç kullanılmasıdır. Üreteçler, şifreleme sistemleri üzerinde anahtarları üreten yapıların ismidir. Bir şifreleme sisteminde anahtarların tahmin edilebilme süresinin zorlaştırılması için, anahtarın rastgele (random) olarak üretilmesi gerekmektedir. Üreteç yönteminde bilinen iki önemli anahtar oluşturma yöntemi bulunmaktadır.

13.4.1 Asimetrik Şifreleme İçin Kullanılan Anahtarlama

Asimetrik şifrelemede bir özel bir de yerel anahtar bulunmaktadır. Yerel anahtar, açık olarak dağıttığımız şifrelerdir. Özel anahtar ise sizin gizli tuttuğunuz şifrelerdir. İki anahtar kullanılan şifreleme algoritmasına göre aralarında bir bağlantı bulunmalıdır. Bu yüzden anahtar üreteçler, açık anahtar şifrelemede kullandıklarında ürettikleri anahtar çiftlerinin, algoritmadaki beklentileri tatmin etmesi gerekir.

Sistemi daha iyi açıklayabilmek için basit bir örnek verelim. Bir kutunuz olduğunu düşünün ve bu kutuyu açabilen yalnızca bir çift anahtar bulunmaktadır. Anahtarlarınızdan birisi herkese açık bir anahtar ve diğeri ise sakladığınız bir anahtardır. Birisi size mesaj gönderdiği zaman herkese açık anahtarınızı kullanarak mesajını şifreleyebilir lakin bu mesajı ancak siz kendinizde bulunan gizli anahtarınız ile açabilirsiniz.

Siz birine mesaj gönderdiğinizde ise (mesajın sizden geldiğinin anlaşılması için) gizli anahtarınızı kullanarak mesajı şifreleyebilir ve herkese açık olan anahtar ile de o mesajı açabilir.

İlk durum mesajın okunmamasını, ikincisi ise mesajın kimden geldiğinin doğrulanmasını sağlamaktadır.

13.4.2 Simetrik Şifreleme İçin Kullanılan Anahtarlama

Simetrik şifrelemede ise anahtar dağıtılmak zorundadır. Yani her iki tarafta anahtarı kullanmak zorundadır. Bu durumda anahtarın iki taraf arasına güvenli bir şekilde iletilmesi problemi ortaya çıkmaktadır. Problemin çözümü için geliştirilen algoritmalara anahtar dağıtma algoritmaları ismi verilir ve bu konudaki en basit algoritmalarından birisi **Diffie-Hellman** algoritmasıdır.

Diğer yöntem ise, bir sonraki şifreyi mesajın içerisine koyarak, mesajı iletme. Fakat her iki yöntemde kendi içerisinde belirli zafiyetleri bulunmaktadır.

13.5 ŞİFRELERİN KIRILMASI

13.5.1 ENİGMA MAKİNESİ

Enigma şifrelemesine karşı tehdit oluşturan saldırıların gerçekleşmesi ile birlikte, **Enigma makinesi**, ilk olarak Birinci Dünya Savaşının sonlarında Alman Mühendis Arthur Scherbius tarafından keşfedilmiştir. Üretilen bu model ve varyasyonları 1920'lerin başlarında ticari amaçlı kullanılmış, ardından en dikkat çeken İkinci Dünya Savaşı öncesinde ve süresince Nazi Almanyası'nda olmak üzere çeşitli ülkelerde askeri ve hükümete bağlı kuruluşlar tarafından benimsenmiştir.

Temelde şifrelenecek metin içerisindeki her karakterin **Vigenere** şifresinde olduğu gibi farklı bir **yer değiştirme şifresi** ile şifrenmesi prensibinde çalışır. Her gün kendi One Time Pad'ini sıfırlamakta ve yeni bir tane üretilmektedir. Makineyi geliştirirken amaçları makinelerin üstünde bir sistem geliştirebilmek ve herhangi bir makinenin yerine geçmesini sağlamaktır.

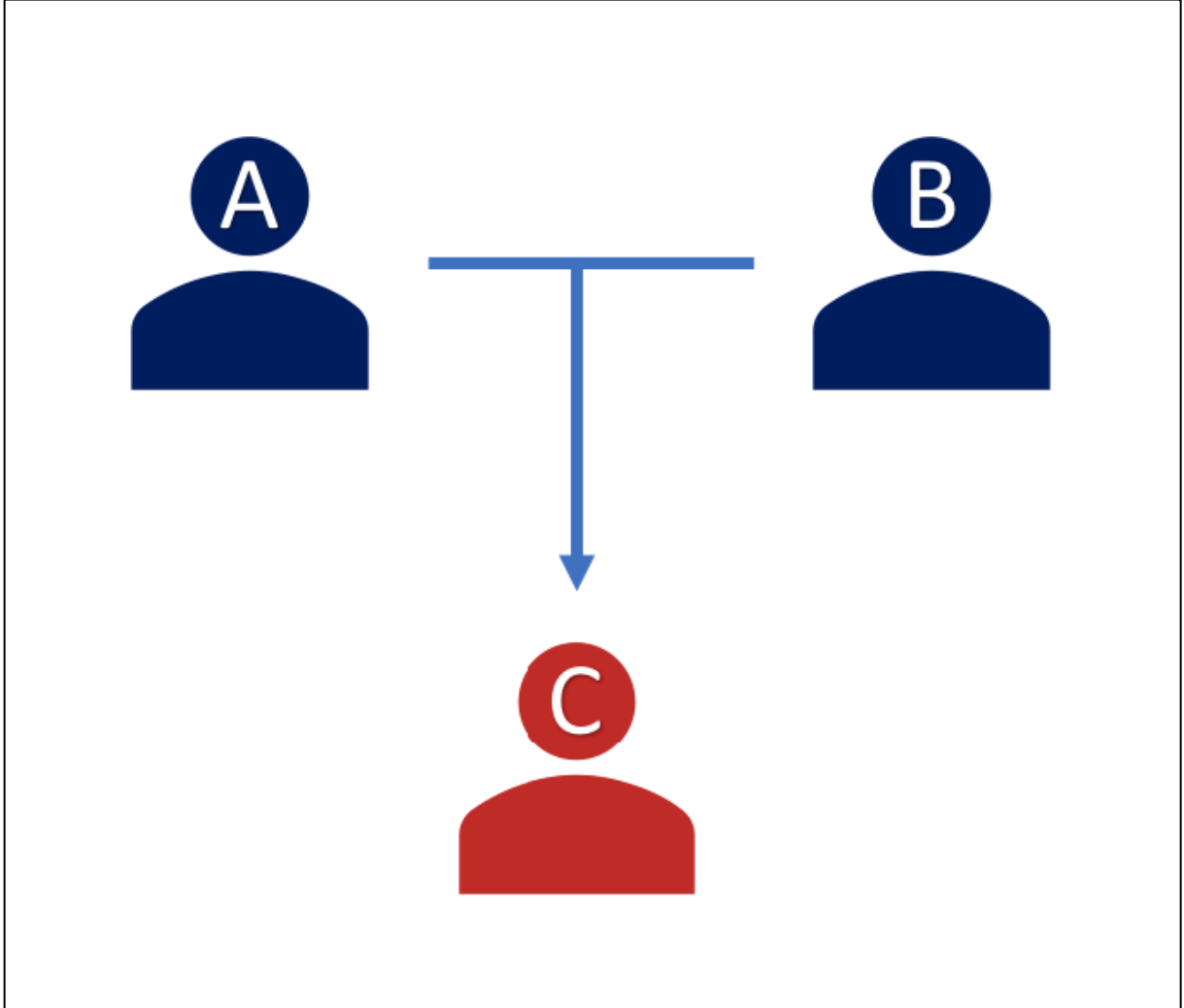


Resim 13.5.1.1

13.5.2 ŞİFRELER NASIL KIRILABİLİR?

Şifre kırılmalarında en çok bilinen yöntemlerden biri olan **Man in the middle (Ortadaki Adam)** yönteminden bahsedilerek açıklanacaktır.

Resim 13.5.2.1’de görüldüğü üzere, A ve B kişileri arasında yer alan iletişimi, başka bir üçüncü şahısın (C) dinlemesi veya mesajlara zarar vermesidir. Burada gerçekleşen saldırılar pasif ve aktif olmak üzere ikiye ayrılmaktadırlar.



Resim 13.5.2.1

Aktif saldırılarda saldırganın olduđu anlaşılmaktadır. Saldırgan kendisini fark ettirmek ister, böylece amacı iletişime daha fazla zarar verebilmektir. Aslı olmayan yerlerden mesajın geldiđini göstermek, mesajı okunamaz hale getirmek ya da mesajın içeriđini deđiřtirmek gibi örnekler verilebilir.

Pasif saldırılarda ise bir saldırganın olup olmadıđı anlaşılamamaktadır. Saldırgan mesajı A kişisinden alabilmektedir ve yine olduđu gibi aktarabilmektedir. Dolayısı ile kullanıcılar C saldırganından haberi olmamaktadır. Ancak burada yer alan problem, saldırganın aldıđı mesajın anahtarına (řifresine) da ulaşabilip, ulaşamayacağıdır.

13.6 ANAHTARLARIN DEĐİŐİŐİMİ

İki kişi arasında anahtarlar deđiřtirilirken (anahtarlar birbirlerine gönderilirken), gönderilen kanal güvenli deđil ise, iletişimi dinleyen 3. bir şahıs anahtarlara erişebilir. Dolayısıyla bu anahtarların tekrardan güncellenmesi gibi bir sorun ortaya çıkacaktır. Anahtarlara erişen saldırgan, kendi anahtarı ile deđiřtirmek gibi birçok istenmeyen duruma sebep olabilir.

Anahtarlara sahip kullanıcıların, anahtarların erişilemeyeceđi şekilde güvenli ortamlarda tutması gerekmektedir. Bu gibi güvenlik açıklarının önüne geçilebilmesi için sertifika sistemleri devreye girmektedir.

13.7 KİŞİSEL ŞİFRELEME (PGP)

13.7.1 PGP NEDİR?

PGP, Phil Zimmermann tarafından geliştirilmiş bir e-mail ve dosya şifreleme protokolüdür. **IDEA, RSA, DSA, MD5, SHA-1** gibi şifreleme algoritmalarını bünyesinde barındırmaktadır. Amacı dosya ve e-maillerin korunmasıdır.

PGP'nin Dünya üzerinde en yaygın kullanılan e-mail şifreleme protokolü olduğu söylenebilir. Bunun sebebi verilerin karşılıklı kontrollerden geçtikten sonra erişilebilir duruma gelmesidir. Açık kaynak kodlu olduğu için her zaman geliştirilebilir durumdadır.

13.7.2 PGP'NİN ÇALIŞMA PRENSİBİ

Farklı bir kullanıcı bize şifreli bir mail göndermek istiyorsa eğer, bizim özel anahtarımıza (Private Key) sahip olmalıdır. Özel anahtarımıza sahip olmadan bize şifreli mesaj yollayamamaktadır.

Diğer bir özelliği ise göndericinin bizim özel anahtarımıza sahip olsa bile, bize gönderilen mesajları okuyamamasıdır (buna kendi gönderdiği mesajlar da dahildir).

Bizim yerel anahtarımızla (Public Key) şifrelenmiş bir postamızı açmanın tek yolu ise özel anahtarımızı bilmesidir.

PGP ile şifreli mail gönderme işlemi şöyle gerçekleşir;

1. Mail gönderecek olan kullanıcı, alıcı tarafından yerel anahtarını talep eder
2. Talep üzerine alıcı yerel anahtarını, alıcı tarafa yollar
3. Yerel anahtara sahip olan gönderici, mailini gönderilen yerel anahtarlar ile şifreler ve alıcıya gönderir
4. Şifrelenmiş mesaj alan alıcı kendisinin özel anahtarı ile şifrelenmiş maili deşifre eder
5. Anlaşıldığı üzere korsanlar gönderilen şifreli maili ele geçirirse bile mail sadece alıcıda bulunan özel anahtar ile açılabileceği için bir faydası olmaz.

Daha önceki kısımlarda da bahsedildiği üzere yerel ve özel anahtarların, her birinin sadece tek bir görevi bulunmaktadır. Yerel anahtar sadece maili şifrelemeye yaramaktadır fakat şifreyi açamamaktadır. Bu nedenle yerel anahtara ulaşılsa bile deşifreleme işlemi gerçekleşmemektedir (PGP'de yer alan asimetrik özelliği bu durum sebebi ile vardır).

PGP'de şifreleme üç adımda gerçekleşmektedir. Birinci adımda gönderen, mesajın özünü oluşturmaktadır.

İkinci adımda mesaj sıkıştırılmaktadır.

Üçüncü ve son adımda ise mesaj şifrelenmektedir.

Alıcı taraf mesajı açmak istediğinde bu adımları tersten uygulamalıdır. PGP şifreleme algoritmaları olarak RSA ve IDEA algoritmaları kullanılmaktadır. Mesajın özü alınırken ise MDS algoritması kullanılır.