1

LiTH, Linköping Teknoloji Üniversitesi

IDA, Bilgisayar Bilimleri Bölümü

Prof. Dr. Christoph Schuba

**Final Sınavı**  **ve**  **Cevap**  **Anahtarı**

**TDTS04/TDTS43 (2007 - vt1)**

**Bilgisayar Ağları**  **ve**  **Dağıtılmış**  **Sistemler**

**15 Mart 2007** **Perşembe**

**8:00** **-** **12:00**  (Linköping)

**Öğrenci Kimliği/adı:**\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| ***Soru #*** | ***Maks.*** ***puanlar*** | ***Gerçek puanlar*** |
| ***1*** | **4** | *1+1+1+1* |
| ***2*** | **8** | *3+2+3* |
| ***3*** | **5** | *1+1+1+2* |
| ***4*** | **6** | *6* |
| ***5*** | **3** | *3* |
| ***6*** | **6** | *4+2* |
| ***7*** | **2** | *2* |
| ***8*** | **6** | *2+1+1+2* |
| ***toplam:*** | ***40*** | ***40*** |

Talimatlar :

1. Adınızı ve kimliğinizi her  sayfaya yazın. Bunu  **şimdi**yap!
2. Bütünlük için bu sınavı kontroledin. Toplam 19  **numaralı** sayfa olmalıdır.
3. Cevaplar İngilizce veya İsveççe olarak verilebilir.
4. Tüm soruları çok dikkatli okuyun ve sadece soruları cevaplayın!

**alakasız**  **metin**!!! **yazarak**    **zaman**  **kaybetmeyin**

1. Kitap, dizüstü bilgisayar, ders notu, komşu vb. aşağıdaki istisnalardışında:
   * English/İsveççe sözlük
   * Temel (programlanamaz) cep hesap makinesi
2. Yanıtlarınızı sağlanan alana yazın. Bu yeterli değilse, ters tarafı veya arkadaki ekstra sayfaları kullanın ve hangi soruyu yanıtladığınızı açıkça belirtin.
3. Yukarıdaki kuralları çiğnemek hile olarak kabul edilecek ve Üniversite politikasına göre ele alınacaktır.
4. **HER** sayfaya adınızı veya öğrenci kimliğinizi yazdınız mı?

# TCP - Tıkanıklık Denetimi (5p.)

1. **TCP**  **yavaş**  **başlatmanın** ne anlama geldiğiniaçıklayın. (1 p.)

**cevap:**

TCP bağlantısı cinsolduğunda, tıkanıklık penceresinin değeri genellikle 1 en büyük segment boyutuna (MSS) başlatılır. Bağlantıya kullanılabilir bant genişliği çok daha büyük olabileceğinden, TCP göndereni, tıkanıklık penceresinin değerini doğrusal olarak artırmak yerine iki katına çıkararak gönderme hızını artırır İlk paket kaybı algılandıktan sonra tıkanıklık penceresi ikiye bölüner ve daha sonra doğrusal olarak büyür.

1. TCP **göndereni üç kat**  **yinelenen**  **ACK** aldığında, tıkanıklık penceresi (CongWin) ikiye bölünr, sonra doğrusal olarak büyür. Ancak, bir TCP göndereni bir **zaman aşımı** olayıyla karşılaştığında, yeni bir yavaş başlangıç çalıştırır.

Gözlemlenen veri kaybının bu farklı tedavisinin arkasındaki felsefeyi açıklayın. (1 p.)

**cevap:**

Trip le duplicate ACKS, ağın en azından bazı segmentler, zaman aşımı ise çok daha endişe verici bir tıkanıklık senaryosu.

Bunedenle, TCP gönderen ikinci senaryoda çok daha agresif bir şekilde geri çekmektedir.

3

1. Explai n, TCP tıkanıklık denetimi bağlamında  **katkı artışı**  ve  **çarpan**  **düşüşü** terimleri . (1 p.)

**cevap:**

TCP, kayıp oluşana kadar kullanılabilir bant genişliği için iletim hızını (örneğin, pencere boyutu) değiştirir. Katkı artışında, TCP, kayıp tespit edilene kadar CongWin'i her RTT'de 1 MSS artırır. Çoklamalı düşüşte,TCP, kayıp gözlendikten sonra CongWin'ini yarıya indiriyor. Bu, bant genişliğiiçin araştırma yaparken testere dişi davranışı ile sonuçlanır.

1. Mümkün olduğunca fazla veri göndermek isteyen iki uygulama (tcp kullanırken, diğeri UDP kullanarak) bant genişliği için  *rekabet* ettiğinde ne olacağını açıklayın. (1 p.)

**cevap:**

TCP kullanılabilir kaynak miktarına uyum sağladığından ve UDP mümkün olduğunca fazla veri gönderdiğinden, TCP bağlantısı geri çekilir ve UDP kullanarak uygulamaya neredeyse tüm ağ bant genişliğinden vazgeçer.

# Yönlendirme (8 p.)

1. Bağlantı Durumu Yönlendirme (3 p.)

Dijkstra algoritmasını kullanarak aşağıdaki tabloyutamamlayın.

A düğümünden tüm ağ düğümlerine giden en kısa yolu hesaplayın.

Not: Olası bağlar en soldaki sütun lehine  **kırılmalıdır.**

B

C

D

ve

A

F

1

1

2

2

2

5

5

5

5

1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***adım*** |  | ***N'*** | ***D(B),p(B)*** | ***D(C),p(C)*** | ***D(D),p(D)*** | ***D(E),p(E)*** | ***D(F),p(F)*** |
| 0 | A |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |

**cevap:**

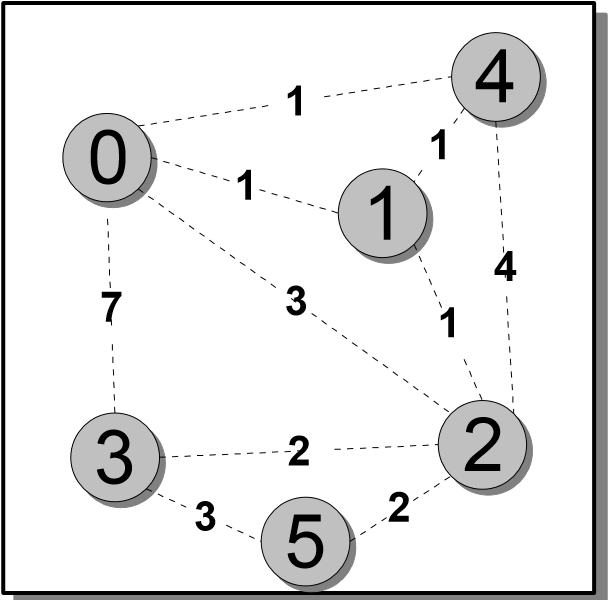
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***adım*** | ***N'*** | ***D(B),p(B)*** | ***D(C),p(C)*** | ***D(D),p(D)*** | ***D(E),p(E)*** | ***D(F),p(F)*** |
| 0 | A | 5, A | 5, A | 5, A | 5, A | 1, A |
| 1 | in | 3, F | 5, A | 5, A | 3, F | yapılmış |
| 2 | Afb | yapılmış | 4, B | 5, A | 3, F |  |
| 3 | AFBE |  | 4, B | 5, A | yapılmış |  |
| 4 | AFBEC |  | yapılmış | 5, A |  |  |

5

1. Mesafe Vektör Yönlendirmesi (2 p.)

Aşağıda gösterilen ağı göz önünde bulundurun ve her düğümün başlangıçta komşularının her birine maliyetleri bildiğini varsayalım. Mesafe vektör algoritmasını kullanın ve algoritma birleştikten sonra düğüm  ***2'de*** like'a bakacağı için aşağıdaki tüm mesafe tablosunu tamamlayın.

Hesaplamanızda Poison Reverse algoritmasını göz önünde bulundurduğunuzdan emin olun!! Bu, bazı girişlerin maliyet ∞ sahip olabileceği anlamına gelir!



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Düğüm***  ***2'deki*** Mesafe ***Vektör***  ***tablosu*** | |  | **C** | **ost'den**  **hedefe**  **no** | | **in** |  |
| ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** |
| **uzaklık**  **Komşudan** vektör | ***0*** |  |  |  |  |  |  |
| ***1*** |  |  |  |  |  |  |
| ***3*** |  |  |  |  |  |  |
| ***4*** |  |  |  |  |  |  |
| ***5*** |  |  |  |  |  |  |

**cevap:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Düğüm***  ***2'deki*** Mesafe ***Vektör***  ***tablosu*** | |  | **C** | **ost'den**  **hedefe**  **no** | | **in** |  |
| ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** |
| **uzaklık**  **Komşudan** vektör | ***0*** | 0 | 1 | 2 | 4 | 1 | 4 |
| ***1*** | 1 | 0 | ∞ | ∞ | 1 | ∞ |
| ***3*** | ∞ | ∞ | ∞ | 0 | ∞ | 3 |
| ***4*** | 1 | 1 | 2 | 4 | 0 | 4 |
| ***5*** | ∞ | ∞ | ∞ | 3 | ∞ | 0 |

1. Karşılaştırma Bağlantısı Durumu ve Mesafe Vektör algoritmaları (3 s.)

Bağlantı durumunu ve mesafe vektör algoritmalarını aşağıdaki kategorilere göre karşılaştırın. Aşağıdakitabloyutamamlayın.

|  |  |  |
| --- | --- | --- |
| ***kriter*** | ***Bağlantı Durumu***  ***Algoritması*** | ***Mesafe Vektör***  ***Algoritması*** |
| ***İleti karmaşıklığı*** |  |  |
| ***Yakınsama hızı*** |  |  |
| ***sağlamlık***  ***( yönlendirici***  ***arızalanırsa*** ***ne***  ***olur?)*** |  | * Herhangi bir düğüm yanlış yol maliyeti bildirebilir * Her düğümün tablosu başkaları tarafından kullanılır. Bu nedenle, hatalar ağ üzerinden yayılır! |

**CEVAP: ( her**  **doğru**  **giriş için**  **1/2**  **puan** **verin.)**

|  |  |  |
| --- | --- | --- |
| ***kriter*** | ***Bağlantı Durumu***  ***Algoritması*** | ***Mesafe Vektör***  ***Algoritması*** |
| ***İleti karmaşıklığı*** | n düğümleri ile, E bağlantıları:O (nE) gönderilen iletiler | * yalnızca komşular arasında alışveriş * yakınsama süresi değişir |
| ***Yakınsama hızı*** | * O(n2) algoritması O(nE) iletileri gerektirir * salınımları olabilir | * yakınsama süresi değişir * yönlendirme döngüleri oluşturabilir * sonsuza kadar sayma sorunu |
| ***sağlamlık***  ***( yönlendirici***  ***arızalanırsa*** ***ne***  ***olur?)*** | * düğüm yanlış bağlantı maliyeti tanıtabilir * her düğüm yalnızca  *kendi* tablosunu hesaplar | * Herhangi bir düğüm yanlış yol maliyeti bildirebilir * Her düğümün tablosu başkaları tarafından kullanılır. Bu nedenle, hatalar ağ üzerinden yayılır! |

7

# Ağ Adresi Çevirisi ve EşlerArası-Eş Bilgi İşlem (17:00)

Alice'in, Kullanıcı adı Bob olan bir eşin indirmek istediği bir dosyaya sahip olduğunu sorgulayarak keşfettiğini varsayalım. Ayrıca Bob'un bir NAT'nin arkasında olduğunu, Alice'in ise olmadığını varsayalım. 138.76.29.7 NAT'nin WANtarafı adresi olsun ve 10.0.0.1 Bob için dahili IP eklemeress olsun. NAT'nin P2P uygulaması için özel olarak yapılandırılmadığını varsayalım.

1. Alice'in eşi, NAT'nin WANtarafı adresi olan 138.76.29.7'yi bilse bile, Alice'in eşinin Bob'un eşine neden TCP bağlantısı başlatamadığını açıklayın. (1 s.)

**cevap:**

Alice Bob ile TCP bağlantısı kurmaya çalıştığında ne olacağını düşünün. Alice, hedef adresi 138.76.29.7 ve bazı hedef bağlantı noktası numarası(örneğin, x) içeren bir TCP SYN paketi gönderebilir. NAT bu TCP SYN paketini aldığında, WAN tarafından başlatılan bir bağlantı için girdisi olmadığından, paketi hangi iç ana bilgisayara yönlendirmesi gerektiğini bilmez. Böylece NAT SYN paketini bırakır.

1. Şimdi Bob'un başka bir eşe devam eden bir TCP bağlantısı kurduğunu varsayalım , Cindy, who bir NAT'nin arkasında değil. Ayrıca Alice'in Cindy'den Bob'un istenen dosyaya sahip olduğunu öğrendiğini ve Alice'in Cindy ile TCP bağlantısı kurabileceğini varsayalım.

Alice'in bu iki TCP bağlantısını (B-C ve A-C) bob'a korkunç bir CT TCP bağlantısı başlatmasını (Cindy'den geçmeden) Alice'e geri döndürmesini öğretmek için nasıl kullanabileceğini açıklayın. (1 p.)

**cevap:**

Alice ve Cindy arasında ve Cindy ile Cindy arasında mevcut bir TCP bağlantısı vardır.

Bob. Bu iki TCP bağlantısıaracılığıyla Alice Bob 'a bir ileti gönderebilir. Özellikle, Alice cbob'dan Alice'e doğrudan TCP bağlantısı başlatmasını isteyin. Bob bu TCP bağlantısını başlattığından, Bob'un NAT'ı aracılığıyla kurulabilir. Alice ve Bob arasında bu doğrudan TCP bağlantısı kurulduktan sonra, Alice Bob'dan dosyayı doğrudan TCP bağlantısı üzerinden göndermesini isteyebilir.

1. Hem Alice hem de Bob'un farklı NAT'lerin arkasında olduğunu varsayalım. Alice'in uygulamaya özgü NAT yapılandırması olmadan Bob ile TCP bağlantısı kurmasına izin verecek bir teknik tasarlamak mümkün mü? Cevabınızı açıklayın. (1 p.)

**cevap:**

Böyle bir teknik tasarlamak mümkün değildir. Alice ve Bob arasında doğrudan TCP bağlantısı kurmak için, Alice veya Bob 'un diğerine bir bağlantı başlatması gerekir. Ama Alice ve Bob kapsayan NAT'ler WAN tarafına gelen SYN paketlerini bırakır. Bu nedenle, her ikisi de NAT'lerin arkasındaysa, ne Alice ne de Bob diğerine TCP bağlantısı başlatamaz.

1. Ağ Adresi Çevirisi teknolojisinin neden hala oldukça tartışmalı olduğu , widely benimsenmiş olmasına rağmen en az iki argüman verin. (2 p.)

**cevap:**

Olası yanıtlar şunlardır:

* + Yönlendiriciler yalnızca katman 3'e kadar işlemelidir - katman ihlali
  + NAT uç-uç bağımsız değişkeniniihlal eder, çünkü uygulama tasarımcıları NAT kutularının katılımcı uygulamalar arasındaki iletişim yolunda olabileceğini dikkate almalıdır.
  + IP adresi sıkıntısı NAT'ler yerine IPv6 ile çözülmeliydi

9

# Bağlantı Katmanı (18:00)

Bağlantı Katmanı tarafından sağlanan kalan altı hizmeti açıklayın. *Çerçeveleme* açıklaması, yanıtlarınızda sağlamanız beklenen ayrıntı düzeyi için bir örnek olarak hizmet eder.

|  |  |
| --- | --- |
| ***Bağlantı Katmanı***  ***Hizmeti*** | ***tarif*** |
| ***Çerçeveleme*** | Hemen hemen tüm bağlantı katmanı protokolleri, bağlantı üzerinden iletimden önce her ağkatmanı verigramını bir bağlantıkatmanı çerçevesinde kapsüller.  Çerçeve, ağkatmanı veri birimi eklendiği bir veri alanından ve bir dizi üstbilgi veya fragman alanından oluşur.  Çerçevenin yapısı bağlantıkatmanı protokolü tarafından tanımlanır. |
| ***Bağlantı erişimi*** |  |
| ***Güvenilir teslimat*** |  |
| ***Akış kontrolü*** |  |

|  |  |
| --- | --- |
| ***Bağlantı Katmanı***  ***Hizmeti*** | ***tarif*** |
| ***Hata algılama*** |  |
| ***Hata düzeltme*** |  |
| ***Yarı çift yönlü ve***  ***tam çift yönlü*** |  |

**cevap:**

|  |  |
| --- | --- |
| ***Bağlantı Katmanı***  ***Hizmeti*** | ***tarif*** |
| ***Çerçeveleme*** | Hemen hemen tüm bağlantı katmanı protokolleri, bağlantı üzerinden iletimden önce her ağkatmanı verigramını bir bağlantıkatmanı çerçevesinde kapsüller.  Çerçeve, ağkatmanı veri birimi eklendiği bir veri alanından ve bir dizi üstbilgi veya fragman alanından oluşur. Çerçevenin yapısı bağlantıkatmanı protokolü tarafından tanımlanır. |
| ***Bağlantı erişimi*** | Orta erişim denetimi (MAC) protokolü, çerçevenin bağlantıya iletildiği kuralları belirtir. Bağlantının bir ucunda tek bir gönderen ve bağlantının diğer ucunda tek bir alıcı olan noktadan noktaya bağlantılar için, MAC protokolü basittir (veya yok)- gönderen bağlantı boşta olduğunda çerçeveyi gönderebilir. Daha ilginç durum, birden çok düğümün tek bir yayın bağlantısını paylaşmasıdır - çoklu erişim sorunu olarak adlandırılır. Burada MAC protokolü birçok düğümün çerçeve transmisyonlarını koordine etmeye yarar . |
| ***Güvenilir teslimat*** | Güvenilir teslimat hizmeti, her ağkatmanı veri birimine bağlantı boyunca hatasız bir şekilde taşımayı garanti eder. Bir taşımakatmanı güvenilir teslimat hizmetine benzer şekilde, bildirimler ve yeniden iletimlerle bağlantıkatmanı güvenilir bir teslimat hizmeti elde edilir.  Bağlantı katmanı güvenilir teslim hizmeti genellikle, verilerin bir aktarım veya uygulama katmanı protokolü tarafından uçtan uca yeniden iletilmesini zorlamak yerine, bir hatayı yerel olarak düzeltmek amacıyla kablosuz bağlantı gibi yüksek hata oranlarına eğilimli bağlantılar için kullanılır. |
| ***Akış kontrolü*** | Bir bağlantının her iki tarafındaki düğümler sınırlı miktarda çerçeve arabelleğe alma kapasitesine sahiptir. Bu olası bir sorundur, çünkü alıcı bir düğüm çerçeveleri işleyebileceğinden daha hızlı bir hızda alabilir. Akış kontrolü olmadan, alıcının arabelleği taşabilir birnd çerçeveleri kaybolabilir. |
| ***Hata algılama*** | Düğümün alıcısı, bir çerçeve olarak iletildiğinde çerçevedeki bir bitin sıfır olduğuna yanlış karar verebilir ve bunun tersi de olabilir. Bu tür bit hataları sinyal zayıflaması veya elektromanyetik gürültü ile ortaya çıkar. Bir hatası olan bir veri birimi iletmeye gerek olmadığından, birçok bağlantıkatmanı protokolü bir veya daha fazla hatanın varlığını algılamak için bir mekanizma sağlar. Bu, iletim düğümünün çerçevede hata algılama bitlerini ayarlaması ve alıcı düğümün bir hata denetimi gerçekleştirmesini sağlayarak yapılır. |
| ***Hata düzeltme*** | Hata düzeltme, hata algılamanın bir adım ötesine geçer, çünkü yalnızca bir hata oluştuğunu algılamaz, ancak çerçevenin tam olarak neresinde oluştuğunu algılar ve sonra bu hataları düzeltin . Bazı protokoller yalnızca paket üstbilgisi/treyler alanları için hata düzeltme sağlar. |
| ***Yarı çift yönlü ve***  ***tam çift yönlü*** | Tam çift yönlü iletimile, bir bağlantının her iki ucundaki düğümler paketleri aynı anda iletebilir. Yarı çift yönlü iletimilebir düğüm aynı anda hem iletemez hem de alamaz. |

# İzleme(15:00)

Traceroute programının nasıl çalıştığını ayrıntılı olarak açıklayın. (15:00)

**cevap:**

Traceroute, gönderilen her ardışık ICMP Yankı İsteği paketinin *canlı*  değerini artırarak çalışır. İlk üç paketin bir time-to-live (TTL) değeri vardır (tek bir atlama yaptıklarını ima eder). Sonraki üç paketin TTL değeri 2'dir ve bu şekilde devam eder. Bir paket bir ana bilgisayar veya yönlendiriciden geçtiğinde, normalde TTL değeri biri tarafından azaltılır ve paket bir sonraki ana bilgisayara iletilir. TTL'si bir ana bilgisayara ulaştığında, ana bilgisayar paketi atar ve gönderene aşılmış bir ICMP süresi (tür 11) paketi gönderir. Traceroute yardımcı programı, paketlerin hedefe giderken geçtiği ana bilgisayarların listesini oluşturmak için bu dönen paketleri kullanır. Traceroute gerçek ana bilgisayarları listelemeyebilir , ilk ana bilgisayarın bir atlamada, ikinci ana bilgisayarın iki atlamada olduğunu gösterir. IP, tüm paketlerin aynı yolu alacağını garanti etmez. Modern Unix ve Linux tabanlı işletim sistemlerinde, traceroute yardımcı programı by varsayılan olarak, kullanımda olması muhtemel olmayan yüksek hedef bağlantı noktası numarasına sahip UDP veri birimlerini kullanır.

# Ağ Güvenliği (18:00)

1. Varlıkların güvenli bir şekildeiletişim kurmasının ne anlama geldiğini sorduğumuz zaman,  *güvenlik*  *hedefleri* olarak adlandırılan bir dizi istenen özellik. Aşağıdaki boşlukları doldurun, beş güvenlik hedefi ile birlikte ne anlama geldiklerinin açıklamasını belirtin. Mümkün olduğunca kesin ol. (16:00)

|  |  |
| --- | --- |
| ***Ağ Güvenliği Hedefleri*** | ***tarif*** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**cevap:**

Not verme kılavuzu:

* + Bu son sınavdan tekrarlanan bir soru olduğu için, örnek girişi kaldırdık .
  + Satırlarda doğru doldurulen ilk iki puan her biri 1/2 puandır.
  + Satırda doğru doldurulen her ek puan her biri 1 puandır.

|  |  |
| --- | --- |
| ***Ağ Güvenliği Hedefleri*** | ***tarif*** |
| ***Gizli -lik*** | Yalnızca gönderenin ve hedeflenen alıcının iletilen bir iletinin içeriğini anlayabilmesi gerektiği kavramı . |
| ***Öz -günlük*** | İletişim kuran tarafların,  iletişim, gerçekten de karşı tarafın kim veya ne olduğunu iddia ettiğini onaylamak için. |
| ***bütünlük*** | Yetkisiz varlıklar tarafından mesajlarda/verilerde herhangi bir değişiklik tespit edilebilir kavramı ; başkabir deyişle, mesajlar tam olarak gönderildikleri gibi alınmalıdır, hiçbir şey eksikolmamalıdır, hiçbir şey eklenmez, hiçbir şey değişmemiştir. |
| ***kullanılabilirlik*** | Kavram, iki veya daha fazla tarafın ilk etapta iletişim kurabilmesi gerektiğini yakalar , örneğin, kaynaklar, örneğin bir ağ hizmeti gibi, erişilebilir olmalıdır. Hizmet Reddi'nin karşılığı. |
| ***Reddedilmeme*** | Bir sözleşmenin - özellikle de İnternet üzerinden kabul edilen bir sözleşmenin- daha sonra ilgili taraflardan biri tarafından reddedilemeyeceğini sağlama kavramı . |

1. İleti özgünlüğü, bütünlük ve gizlilik. (2 p.)

Hem Alice hem de Bob'un kendi ortak/özel anahtar çiftleriolduğunu varsayalım. Ayrıca, her ikisinin de birbirini tanıdığını varsayalım ' ortak anahtarları düzeltin ve özel anahtarlarından ödün verilmediğini .

Şimdi,Alice Bob'a bir mesaj göndermek istiyor ve  orijinalliğinin, bütünlüğününve gizliliğinin güvence altına olduğundan emin olmak istiyor.

Alice Bob 'a  *{{m}K+B}K-A* gönderir. ( Yani, sonucu Bob'a göndermeden önce m'yi Bob'un ortak anahtarıyla şifreler , sonra kendi özel anahtarıyla.)

Bu yaklaşımın Alice'ın belirtilen güvenlik hedeflerini gerçekleştirip gerçekleştirmedığını açıklayın.

Bunu yapmazsa, daha iyi bir yaklaşım önerin ve neden daha iyi olduğunu açıklayın.

**cevap:**

Verilen protokol iletinin özgünlüğünü korumaz , çünkü Alice'in ortak anahtarına sahip olan herkes, Trudy de dahil olmak üzere iletiden dış şifreleme katmanını kaldırabilir. Trudy daha sonra sonucu farklı bir özel/ortak anahtar çifti ile yeniden şifreleyebilir. Mesaj o zaman Trudy'den kaynaklanmış gibi görünürdü, Alice'den değil.

Sorunu düzeltmek için, iki encryption adımını ters sırada uygulamanız yeterlidir.

Yani., önce Alice'in özel anahtarıyla  *m* şifrele, sonra Bob'un ortak anahtarıyla. Bu çok daha iyi bir yaklaşım, çünkü sadece Bob şifrelemenin dış katmanını açabilir .

Bu nedenle Alice Bob 'a  *{{m}K-A}K+B* gönderir.

# CORBA (14:00)

Patent buluşu açıklamalarını elektronikolarak göndermek için istihdamları tarafından kullanılabilecek bir müşteri-sunucu CORBA programı yazmak için yüksek teknoloji şirketinin hukuk departmanı tarafından danışman olarak işe aldığınızı düşünün. Bu sys tem'in bir parçası olarak, belirtilen ortak yazarların gerçekten şirketin geçerli olup olmadığını belirleyen bir rutin için bir arabirim belirtmeniz istenir. Aşağıdakileri bilmeniz gerekir:

* Çalışanlar altı haneli çalışan kimlikleri ile benzersiz bir şekilde tanımlanır (örneğin,"AB0001")
* Gönderen yazarın birincil yazar olarak tanımlanması gerekir
* Toplamda kaç ortak yazar olduğunun belirtilmesi gerekiyor.
* Her ortak yazarın çalışan kimliği belirtilmelidir
* Buluşun ifşası için bir ism gerekiyor
* Yordam, gönderilen her çalışan kimliği listesinden geçer, çalışan kimliğin insan kaynakları veritabanında olduğunu doğrulamaya çalışır ve arama başarılı olup olmadığını döndürür.
* Son olarak, tüm ortak yazarlar insan resources veritabanında başarıyla bulunduysa, rutin yeni buluş açıklaması için benzersiz bir tanımlayıcı döndürür ( örneğin,"MSC07-0539")

Boşlukları doldurarak önceki istemci-sunucu etkileşimi için IDL'de **(InventionDisclosure**.idldosyası) bir arabirim açıklaması belirtin. Bize en uygun veri türleri (aşağıdaki tabloya bakın) ve ne anlama geldiğini açıkça ortaya koyarak bağımsız değişken adları kullanın. (2 p.)

arabirim InventionDisclosure

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ checkAuthors(\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_); }

|  |  |  |  |
| --- | --- | --- | --- |
| **ıdl** | **C++** | **ıdl** | **C++** |
| Boolean | Bool | çift | çift |
| Char | imzalı karakter | herhangi | ( yerel eşdeğeriyok , void'e benzer \*) |
| Sekizli | 8 bit | nesne | ( sınıfa en yakın ) |
| numaralandırma | numaralandırma | dizgi | sınıf dizesi (sınırlanmış veya sınırsız uzunluk) |
| kısa | kısa | Yapı | yapı (C++yerine C gibi) |
| imzalanmamış kısa | imzalanmamış kısa | sendika | birleşim (C yapı türü etiketli) |
| uzun | uzun | dizi | [ ] |
| imzalanmamış uzun | imzalanmamış uzun | sıra | ( sınırlanmış veya sınırsız uzunlukta parametrize edilmiş bir dizi ) |
| yüzmek | yüzmek |  |  |

**cevap:**

Not verme kılavuzu: 2 p ile başlayın. 1/2 p çıkarın. dönüş türündeki hatalar için, ilk üç dize bağımsız değişkeni, dize dizisi ve boole dizisi.

arabirim InventionDisclosure

{

dize checkAuthors (dize primaryAuthor, dize numarasıAuthors, dize inventionTitle, dize dizisi authorIDs, dışarı boole dizisi successFlags);

}

# Dağıtılmış Koordinasyon Tabanlı Sistemler (18:00)

1. Koordinasyon modellerinin taksonomisi (2 s.)

Gölgeli tablo girişlerini doldurarak aşağıdaki taksonomiyi tamamlayın.

**zamansal**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | posta kutusu |
| **Ayrılmamış** |  |  |

**Başvuru**

**cevap:**

**zamansal**

|  |  |  |
| --- | --- | --- |
|  | **Birleştiğinde** | **Ayrılmamış** |
| **Birleştiğinde** | dolaysız | posta kutusu |
| **Ayrılmamış** | Toplantı odaklı | Generatif  iletişim |

**Başvuru**

1. Üretken iletişimde anahtar fikir nedir? (1 p.)

**cevap:**

Üretken iletişimde anahtar fikir, bağımsız süreçler koleksiyonunun paylaşılan kalıcı bir dizi veri alanını kullanmasıdır. İşlemler paylaşılan veri alanına her türlü kaydı koyabilir ve ilişkilendirilebilir bir arama mechanism aracılığıyla dizileri bulabilir.

1. Leasing adlı  mekanizmayı açıklayın. (1 p.)

**cevap:**

Leasing nesnelere yapılan başvuruları yönetmek için bir mekanizmadır. Leasing'de, başvurulan nesneler, kendilerine kimin atıfta bulunup, referans listeleri olarak bilinenlere yol açan . Bu liste empty olduğunda, nesne kendini güvenle yok edebilir .

1. JavaSpaces Uygulaması (2 p.)

Güvenilir yayının kullanılabildiği çok bilgisayarlı bir sisteminiz olduğunu varsayalım.

Uygulama mimarisi için ciddi bir aday, tüm alt alanları tüm makinelerde tam olarak çoğaltmaktır. WRITE, READ ve TAKE işlemleri için ne olması gerektiğini açıklayın. Yanıtlarınızı göstermek için, açıklamalarınıza diyagramlar ekleyebilirsiniz.

YAZMA işlemi:

**CEVAP**:

Bir WRITE işlemi yapıldığında, yeni dizi örneği yayınlanır ve her makinede uygun altuzaya girilir.

OKUMA işlemi:

**cevap:**

OKUMA işlemi yapmak için yerel alt alan aranır. Öğe varsa,okunabilir.

TAKE işlemi:

**cevap:**

Bir READ işlemine benzer şekilde, yerel alt alan aranır. Ancak, bir TAKE işleminin başarıyla tamamlanması, dizi örneğinin JavaSpace'ten kaldırılmasını gerektirdiğinden, tüm makinelerden kaldırmak için bir silme protokolü gerekir. Yarış koşullarını ve deadlocks önlemek için, iki fazlı bir tamamlama protokolü kullanılabilir.

Not: İllüstrasyonlar için [TvS2002] ders kitabında Şekil 12-17'ye bakın.

Not verme kılavuzu: 1/2 p. her biri YAZMA ve OKUMA işlemleriiçin doğru açıklamalar için . 1 p. TAKE işlemi için.