

Bulut-Vekil Sunucu ve Tabletler Arasında Veri Senkronizasyonu için RTT Kullanarak Farklı Kümeleme Metotlarının Değerlendirilmesi

Evaluation of RTT based Clustering Methods for Data Synchronization between Proxy-Cloud Servers and Tablets

Süleyman Eken¹, Fidan Kaya¹, Ahmet Güngör¹, Ahmet Sayar¹, Adnan Kavak¹, Suhap Şahin¹

¹Bilgisayar Mühendisliği
Kocaeli Üniversitesi

{suleyman.eken, fidan.kaya, ahmet.gungor, ahmet.sayar, akavak, suhapsahin}@kocaeli.edu.tr

Özet

Veri kopyalanarak farklı amaçlar için coğrafi olarak dağıtık depolama birimlerinde tutulur. Aynı verilerin farklı lokasyonlarda tutulması beraberinde veri tutarsızlığı problemini getirmektedir. Bu problem farklı veri senkronizasyonu teknikleri ile giderilebilir. Veri senkronizasyonunu başlatmak için uygun zaman aralıklarının belirlenmesi ağ bant genişliği ve haberleşme birimlerinin erişilebilirliği gibi birçok parametreye bağlıdır. Bu çalışmada, Fatih projesindeki yerel ve bulut sunucular arasındaki veri senkronizasyonu için RTT tabanlı kümeleme metotları değerlendirilmiştir. Önerilen teknik büyük veri setlerinin transferi için ağın erişilebilirliğini araştırarak diğer sistemlere de uygulanabilir.

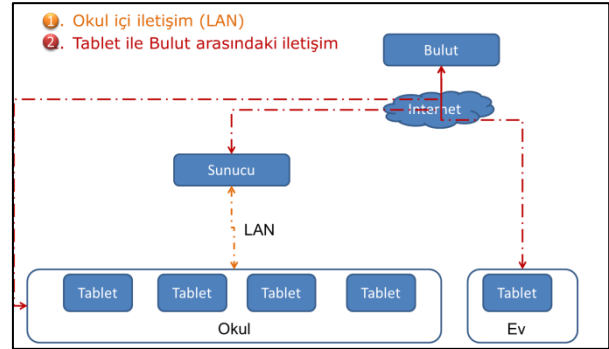
Abstract

Data are replicated and kept on geographically dispersed storages for different purposes. However, keeping the same data on different storages might cause data inconsistency problems. This type of problems can be solved by using data synchronization techniques. However, determining appropriate time intervals to start data synchronization depends on many factors such as network bandwidth and availabilities of communicating peers. The study presented in this paper evaluates RTT based clustering methods for data synchronization between proxy and cloud servers in Fatih project. The proposed technique can also be applicable to any other system scrutinizing availability of the network for transferring relatively big data sets.

1. Giriş

İnternet teknolojilerinin gelişimi ile birlikte kullanılan veri miktarı da akıl almaz bir şekilde artış göstermektedir. Örneğin Fatih (Fırsatları Arttırma ve Teknolojiyi İyileştirme Hareketi) projesi ile beraber öğrenci ve öğretmenlerde bulunan tabletlerden kaynaklı müthiş bir veri boyutu artışı ve trafiği meydana gelmektedir. Bu tabletler ile eğitimsel verinin saklandığı ve bu verilerin dağıtımının sağlandığı bulut sunucular arasında ağ trafiği ve altyapıdan kaynaklanan kısıtlamalar mevcuttur. Bu trafik verilerin etkin bir şekilde transferi konusunda sorunlara neden olmaktadır. Dolayısıyla okul ağında bulunan bir tabletin bulut ortamından ulaşmak istediği eğitimsel veriye ulaşım süresinin ve maliyetinin artmasına neden olabilmektedir. Bu sıkıntıların üstesinden gelmek için

daha önceki çalışmalarımızda okul düzeyinde “yerel bulut sunucu” kullanılmasına ve bu sunucunun ana bulut sistemi ile entegrasyonuna dayalı sistem mimarisi önerilmiştir [1]. Bu çözüm önerisine göre okul ağı içinde bulunan bir tablet yerel sunucu üzerinden bulut sunucu ile haberleşirken okul dışında bulunan bir tablet direk olarak bulut sunucu ile haberleşebilmektedir (Şekil 1’e bakınız). Böylelikle okul ortamında içerik transfer trafiği optimize edilmeye çalışılmıştır.



Şekil 1: Vekil sunucu tabanlı içerik saklama ve dağıtım sistemi

Aynı verilerin farklı yerlerde (tablet-vekil sunucu-bulut sunucu) tutulması verilerin tutarsız olması problemini yanında getirmektedir. Bu problem verilerin senkronize edilmesi ile çözülmektedir. Literatürde senkronizasyon işlemi için farklı yöntemler, yazılımlar ve araçlar önerilmiştir. Daha önceki çalışmamızda senkronizasyon işlemi, NoSQL veri tabanı çeşitlerinden doküman tabanlı olan CouchDB’nin kendi içinde var olan replikasyon protokolü ile gerçekleştirdik [2]. Eğer sistem aktörleri arasındaki senkronizasyon işlemi çok sık yapılırsa ağ trafiği problemi devam edecektir. Ön tanımlı zamanlarda bu işlemi yapmak ise sistemin yoğunluğuna göre karar verme mekanizmasına ters olacağından senkronizasyon işlemi, sistemin veri alış-verişi yapmadığı (veya daha az yaptığı) zamanlarda yapmak daha efektif olmaktadır. Yani kısaca sistem yoğun olmadığı (boşta olduğu) zaman dilimlerinde senkronize edilmesi gerekmektedir. Daha önceki çalışmamızda bu işlemi, ağa gelen/giden paketleri sistem kaynaklarına (CPU ve bellek kullanımı) göre modelleyerek yaptırmiştık [3]. Bu çalışmamızda ise sistemin boşta olma

durumunu, paketlerin gidiş/geliş gecikmelerine (RTT/Round Trip Time) göre farklı kümeleme yöntemleri altında sınıflandırarak araştırmaktayız.

Makalenin geri kalanı şu şekilde organize edilmiştir: 2. bölümde ağ karakteristikleri sunulacaktır. 3. bölümde literatürde var olan ağ trafiği ölçme yöntemleri irdelenmiştir. Sonraki bölümde, veri madenciliği sınıflandırma ve kümeleme yöntemleri üzerinde durulmuştur. 5. bölümde en uygun senkronizasyon zamanını farklı kümeleme yöntemleri ile karşılaştırarak bulunmasına yönelik önerdiğimiz sistemin mimarisi açıklanmıştır. Son bölümde ise sonuçlar değerlendirilmiş ve ileride ne gibi çalışmalar yapılacağından bahsedilmiştir.

2. Ağ Karakteristikleri

Uygun senkronizasyon zamanının bulunması ağ karakteristiklerinin bilinmesini gerektirmektedir. Ağ ölçüm çalışma grubu (Network Measurement Working Group) farklı ağ karakteristikleri tanımlamışlardır. Bu metrikler aşağıdaki alt bölümlerde detaylı olarak açıklanmıştır [4].

2.1. Bant Genişliği Kapasitesi (Bandwidth Capacity)

Ağın kapasitesi bant genişliği ile ifade edilir. Başka bir ifadeyle bir kanaldaki taşınabilecek en fazla frekansa sahip sinyaldır. Burada hiç trafik olmadığı kabul edilir. Bant genişliği ne kadar büyükse, birim zamandaki aktarılabilen veri de o kadar büyük olur. Bant genişliği, verinin hacmiyle doğru orantılıdır. İletişim altyapısının niteliğini bant genişliği belirler. Dolayısıyla işin ekonomik boyutu da bulunmaktadır. LAN bağlantılarında 10, 16, 100, 150, 622 Mbps ve 1 Gbps gibi yüksek hızlardan bahsedilirken, WAN bağlantılarında 33.6K bps, 56Kbps ve 2Mbps gibi daha düşük hızlardan bahsedilir. Bant genişliği kavramı network hızı için önemli bir kavram olmasına rağmen çoğu zaman karıştırılır.

2.2. Bant Genişliği Kullanımı (Bandwidth Utilization)

Bant genişliği kullanımı, güncel olarak kullanılmakta olan bant genişliğini ifade etmektedir. Trafik tipi ve bant genişliği kullanımı hakkında bir bilgi olmadan uygulamalar arasında bant genişliğinin düzgün dağılımını sağlamak mümkün değildir. Video streaming ve torrent vb gibi uygulamalar aşırı bant genişliği kullanımı yaparlar ve kritik uygulamalar için yeterli bant genişliği kalmayabilir. Bunu önceden düşünüp önlemini almak gerekir. Bant genişliği kullanımını izleyerek (monitoring) kritik bant genişliği belirlenebilir.

2.3. Kullanılabilir Bant Genişliği (Available Bandwidth)

Bant genişliği, saniyedeki bit miktarı (bps) ile ölçülür. Yani birim zamanda bir yerden başka bir yere birim zamanda geçen bilgi miktarıdır. Bant genişliği kapasitesinden kullanılan bant genişliği çıkartılarak bulunabilir. Örneğin bant genişliğini bir şehrin su şebekesine benzetirsek; var olan boru genişliği kapasiteyi verirken, kullanımda olan kısım kullanılan bant genişliğini, geriye kalan kısım ise mevcut kullanılabilir bant genişliğini ifade etmektedir.

2.4. Verimlilik (Throughput)

Bilgisayar teknolojisinde “throughput” verilen belirli bir zaman periyodunda bilgisayarın yaptığı işlem hacmi olarak

tanımlanabilir. Bir ağ linkinin bazı kısıtlar altında bir uygulamaya birim zamanda sağladığı maksimum bit miktarının ölçüsü olarak düşünülebilir.

2.5. Paket Gecikmesi (Packet Delay)

Bir paket ağda dolaşırken her bir subnet linkindeki gecikmelerin toplamı paket gecikmesini vermektedir. İki tür gecikme vardır: bir yönlü ve iki yönlü (Round Trip Time-RTT). Bir paketin göndericiden alıcıya gitmesi sırasında yaşanan gecikme tek yönlü gecikmeyi ifade ederken; gönderilen pakete alıcıdan ACK bilgisinin de gelmesine kadar geçen süre çift yönlü gecikmeyi anlatır. Ayrıca havanın sıcak olması, yönlendirilerin yavaş çalışması gibi fiziksel etkenler de yavaşlığa sebep olur.

2.6. Kayıp Paket Sayısı (Count of loss packets)

Bilgisayarlar arası iletişimde veriler “paketler” hâlinde iletilir ve alınır. Tek bir paket bile eksik gitse bu veri “anlamsız veri” olmuş olur. Paket kaybı veri aktarımları sırasında ortaya çıkan tam olarak gönderilemeyen veya gönderim sırasında ortaya çıkan hata durumları olarak tanımlanır. Paket kaybına sebep olan başlıca etmen verimlilik performansdır; ancak bunun yanında farklı kayıp sebepleri vardır.

3. İlgili Çalışmalar

Bölüm 2’de verilen ağ karakteristiklerini kullanan çeşitli ağ yönetimi, özellikle ağ trafiği ölçüm mekanizmaları önerilmiştir. Ağ trafiği ölçümünde haberleşme ağındaki trafiğin tipi ve miktarı ile ilgilenilir [5]. Efektif bant genişliği yönetimi açısından ağ trafiği ölçümü hayati bir işlemdir.

Literatürde bant genişliği ölçüm ve tahmini için farklı yöntemler önerilmiştir ve bunlar sınıflandırılmıştır. Bunlardan ilki aktif ve pasif ölçümdür. Aktif ölçüm yapan araçlar aktif olarak ağa deneme paketleri (probing packets) gönderirken pasif araçlar ağa müdahale etmeden işlem yaparlar [6]. İkinci sınıflandırma tekniği uçtan uca (end-to-end) veya atlamadan atlamaya (hop-by-hop) olup olmadığıdır. End-to-end tekniğinde [7], [8] yol kapasitesi ve kullanılabilir bant genişliğine bakılırken hop-by-hop tekniğinde ise [9], [10] link kullanımı ve kapasitesine bakılır. End-to-end bant genişliği, hop-by-hop bant genişliği ölçümleri bilinirse bundan hesaplanabilir [11]. Üçüncü sınıflandırma olarak SNMP-tabanlı (Basit ağ yönetim protokolü) ve aktif izlemeli düşünülebilir. MTRG [12] gibi SNMP-yardımlı araçlar bireysel linklerin kullanımını izlemek için kullanılır. Aktif izleme metodları ağ içerisindeki en az 2 bitiş noktası arasındaki ölçümleri toplamak için Internet Kontrol Mesaj Protokolü paketleri (Internet Control Message Protocol) gönderirler [13]. Dördüncü olarak ölçülen metrik tipleri ağa müdahale eden ve ağ yanlısı gibi düşünülebilir. “Iperf” [14] ve “ttcp” (Test TCP) [15] gibi ağ müdahaleci ölçüm araçları ağ ekstra bir yük getirmektedir. Diğer yandan ağ yanlısı teknikler bant genişliğini ağa ekstra bir yük getirmeden tahmin ederler.

Ayrıca literatürde ağı karakterize eden bazı araçlar ve yazılımlar vardır. Bunlardan bazıları: Cprobe [16] uçtan uca mevcut bant genişliğini ölçmek için önerilmiş ilk araçtır. Bu metod araştırma için paket çiftini kullanır. Manish ve Constantinos tarafından önerilen Pathload ise uçtan uca mevcut bant genişliğini tahmin eder [17]. Yaptıkları çalışmayı path yönlendiricilerden elde ettikleri veriler üzerinde SNMP

ile karşılaştırmışlardır. Pchar [18], Internet boyunca uçtan uca linkler arasında bant genişliğini, gecikmeyi ve paket kaybını ölçen bir araçtır [19]. BitMeter OS yazılımı [20], kullanıcının seçtiği bir zaman aralığında gerçek zamanlı bant genişliğini verisini, kullanıcının istediği zaman aralığına göre gerçek gösterir. Bu yazılımlar ve araçlar karşılaştırıldığında önerdiğimiz yaklaşım açık kaynaklar kullanarak geliştirilmiştir ve kullanıcıların ihtiyacına göre şekillendirilebilir.

4. Sınıflandırma ve Kümeleme Yöntemleri

Sınıflandırma yöntemleri veri madenciliğinde en çok kullanılan yöntemlerdendir. Sınıflandırma için en çok kullanılan yöntemler arasında; C4.5 gibi karar ağaçları (decision trees), k en-yakın komşu gibi örnek tabanlı yöntemler (instance based methods), Bayes sınıflandırıcı (Bayes classifier) lojistik regresyon (logistic regression), diskriminant analizi (discriminant analysis), yapay sinir ağları (artificial neural networks), bulanık setler (fuzzy sets) ve genetik algoritmalar (genetic algorithms) yer almaktadır [21]. İnsanlar verileri daima sınıflandırdıkları, kategorize ettikleri ve derecelendirdikleri için sınıflandırma, hem veri madenciliğinin temeli olarak hem de veri hazırlama aracı olarak da kullanılabilir [22].

Genel olarak sınıflandırma eğitici (supervised) ve eğitici (unsupervised) olarak ikiye ayrılır. Eğitici sınıflandırmada dışarıdan bir eğitimin müdahalesi söz konusudur. Eğitici, sisteme ilgili girdi için üretmesi gereken sonucu verir. Yani sisteme girdi/çıktı ikilisinden oluşan örnekler sunulur. Eğitici sınıflandırma da ise hiç bir eğitime ihtiyaç yoktur. Tahmin edici, kendine gösterilen örnekleri alır ve belli bir kriter gere göre sınıflandırır. Bu kriter önceden bilinmeyebilir. Sistem, kendi öğrenme kriterlerini kendisi oluşturmaktadır. Kümeleme ise eğitici öğrenme ile gerçekleştirilir. Eğitici ve eğitici sınıflandırma türleri bazen birbirini destekleyici olarak kullanılırlar. Eğitici tür ile özellik uzayı kümelenir. Bu kümeler Eğitici tür için bilinen yapı olarak eğitim kümesinde kullanılabilir. Böylece özellik uzayının boyutu azaltılır. Fakat bu işlemler zaman aldığından gerçek zaman uygulamaları zor ve pahalıdır [23].

Bu çalışmada; Bayes karar kuralı sınıflandırıcısı, C4.5 algoritması ve K-Means algoritması kullanılarak RTT değerleri sınıflandırarak sonuçları karşılaştırılmıştır.

4.1. Bayes Sınıflandırıcı

Bayes teorisi sınıflandırma işlemine bir olasılık problemi gibi yaklaşmaktadır. Naive Bayes yöntemi sınıflandırma işlemlerinde en sık kullanılan bir sınıflayıcıdır. Diğer bütün sınıflandırıcılarla karşılaştırıldıklarında en düşük hata oranına sahiptirler. Elimizde n adet sınıf olduğunu farz edelim, $S_1, S_2, \dots, S_n$. Herhangi bir sınıfa ait olmayan bir veri örneği X 'in, hangi sınıfa ait olduğu Naive Bayes sınıflandırıcı tarafından belirlenir. Veri örneği X , verilen sınıflara ait olma olasılığı en yüksek değere sahip sınıfa atanır. Sonuç olarak, Naive Bayes sınıflandırıcı bilinmeyen örnek X 'i, S_i sınıfına atar. Her veri örneği, m boyutlu özellik vektörleri ile gösterilir, $X = (X_1, X_2, \dots, X_m)$. Özelliklerin hepsi aynı derecede önemlidir ve birbirinden bağımsızdır. Bir özelliğin değeri başka bir özellik değeri hakkında bilgi içermez. $X = (X_1, X_2, \dots, X_m)$ örneğinin S_i sınıfında olma olasılığı (1)'deki gibidir.

$$P(S_i|X) = \frac{P(X|S_i)P(S_i)}{P(X)} \quad (1)$$

$P(X)$ bütün sınıflar için sabit ise, X örneğinin S_i sınıfında olma olasılığına, $P(X|S_i) P(S_i)$ ifadesi ile ulaşabilir. $P(S_i)$, her bir sınıfın olasılığı olup (2)'teki gibidir.

$$P(S_i) = \frac{O_i}{O} \quad (2)$$

Burada O_i , S_i sınıfına ait eğitilen örnek sayısı ve O ise toplam eğitilen örnek sayısıdır. Eğer sınıf öncelik olasılığı bilinmiyorsa, o zaman genel olarak sınıflar eşit kabul edilir, $P(S_1) = P(S_2) = \dots = P(S_n)$, ve bu sebeple $P(X|S_i)$ ifadesi, X örneğinin S_i sınıfında olma olasılığını bulmak için kullanılır. Aksi takdirde, $P(X|S_i) P(S_i)$ ifadesi bizim için en anlamlı ifadedir. Olasılıklar $P(X_1|S_i)$, $P(X_2|S_i)$, ..., $P(X_m|S_i)$ eğitim örneklerinde tahmin edilebilirler,

$$P(X_k|S_i) = \frac{O_{ik}}{O_i} \quad (3)$$

Eşitlik (3)'te O_{ik} , X_k değerine sahip olan S_i sınıfına ait eğitim seti sayısı, ve O_i de S_i 'ye ait olan eğitim seti sayısıdır. Bilinmeyen örnek X 'i sınıflandırmak için, her S_i sınıfı $P(X|S_i) P(S_i)$ ifadesi hesaplanır. Örnek X 'i en yüksek değere sahip S_i sınıfına atanır.

$$P(X|S_i) = \prod_{k=1}^m P(X_k|S_i) \quad (4)$$

4.2. C4.5 Algoritması

80'li yılların başında ID3 (Iterative Dichotomiser) diye isimlendirilen karar ağacı algoritması J. Ross Quinlan tarafından önerilmiştir [24]. Bu yöntem daha sonra E.B. Hunt, J. Marin, and P. T. Stone tarafından önerilen eğitim sistemine göre geliştirilmiştir [25]. Quinlan tarafından C4.5 algoritması ID3 algoritmasına göre yeniden düzenlenmiştir. C4.5 ağacının ID3 ağacından en büyük farkı normalleştirme (normalization) kullanıyor olmasıdır. Yani ID3 ağacı üzerinde entropi hesabı yapılır (veya bilgi kazanımı) ve bu değere göre karar noktaları belirlenir. C4.5 ağacında ise entropi değerleri birer oran olarak tutulur. Ayrıca ağaç üzerinde erişim sıklıklarına göre alt ağaçların (subtree) farklı seviyelere taşınması da mümkündür. C4.5 ağacının diğer bir farkı budama (pruning) işlemi yapılmaktadır.

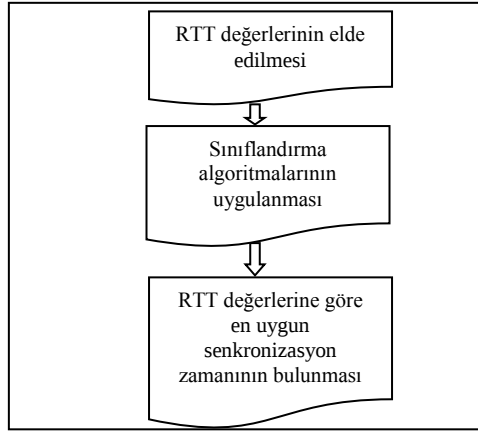
C4.5 algoritması birkaç temel duruma sahiptir: (i) veri集中的 her bir örnek aynı sınıfa ait ise entropi değeri sıfır olmakta ve tek bir düğümden ağaç oluşmaktadır. (ii) özelliklerden hiçbir bilgi kazancı sağlamazsa yeni bir karar düğümü oluşturulur. (iii) Daha önce rastlanmayan bir sınıfa karşılaşıldığında yeni bir karar düğümü oluşturulur. Algoritma genel olarak ise ilk adımda yukarıdaki temel durumları öncelikle kontrol eder. Daha sonra her bir özellik için normalize edilmiş bilgi kazancı oranı hesaplanır. En yüksek değerli kazancı göre karar ağacı oluşturulur.

4.3. K-Means Algoritması

K-Means algoritması en basit eğitici öğrenme algoritmalarından bir tanesidir. Sınıflandırma problemlerini çözme işlemlerinde kullanılır. K-means algoritması ile mevcut olan veriler özelliklerine göre k-adet farklı sınıfa bölünebilir.

Gruplama işlemi küme merkezlerinin sürekli olarak hesaplanması ve kümelerin oluşturulması mantığına dayanmaktadır. Sınıflar oluşturulduktan sonra her sınıf etiketlenir ve algoritma tamamlanmış olur.

5. Sistem Mimarisi



Şekil 2: Önerilen sistemin mimarisi

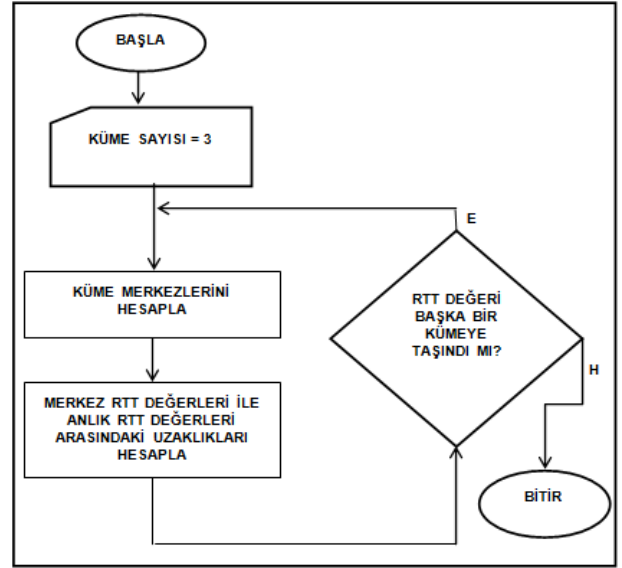
RTT temel olarak bir paketin alıcıya gönderilmesinden, ACK (alındı mesajı) alınmasına kadar geçen süre olarak tanımlanabilir. Aradaki bu süre sinyalin iki uç arasındaki yayılım zamanına bağlıdır [26]. TCP olarak kurulan bir bağlantıya ait RTT değerinin hesaplanmasına ait formül aşağıda verilmiştir.

$$RTT = (\alpha * Eski_RTT) + ((1-\alpha) * Yeni_RTT_Örneği) \quad (5)$$

Çalışma kapsamında oluşturulan modelde bir gün boyunca bir dakikalık periyotlarla ortalama RTT değerleri hesaplanmıştır ve bu değerler kaydedilmiştir. RTT değerlerinin saklanmış olduğu doküman günlük olarak yenilenen bir dokümandır. Bu dokümana göre gün boyu kullanılacak yoğunluk sınıfları her gün yeniden hesaplanmaktadır. Sınıflandırma işlemi ile ilgili detaylı bilgi aşağıdaki alt bölümlerde verilmektedir. Genel olarak önerilen sistemin mimarisi Şekil 2’de verildiği gibidir.

5.1. K-Means Algoritması ile Kümelerin Belirlenmesi

K-Means algoritmasına ait akış diyagramı Şekil 3’te gösterilmiştir. Kullanılacak sınıf sayısı üç (senkronizasyon için uygun/düşük, nötr/orta ve yoğun/yüksek) olarak belirlenmiştir. Algoritmanın başında ilk olarak küme merkezleri tanımlanmıştır. Birinci kümenin merkezi RTT dokümanında yer alan en küçük RTT değeri olarak belirlenmiştir. İkinci kümenin merkezi RTT dokümanında yer alan en büyük RTT değeri olarak belirlenmiştir ve üçüncü kümenin merkezi ise RTT dokümanında yer alan RTT değerlerinin ortalaması alınarak belirlenmiştir. Bu merkez değerlerine göre kümeler oluşturulmuştur.



Şekil 3: K-Means algoritması adımları

K-Means algoritması ile bütün RTT değerleri bir kümeye dahil edildikten sonra anlık bir RTT değeri hangi kümeye daha yakın ise ona göre işlem yapılmaktadır.

5.2. Bayes Sınıflandırıcı ve C4.5 ile Sınıfların Belirlenmesi

Hem Bayes hem de C4.5 algoritmalarının kullanılabilmesi için veri setinde var olan tek özellik (RTT) yanına sınıflandırma etiketinin de eklenmesi gerekmektedir. Bu özellik K-Means algoritmasından elde edilen nihai küme merkezlerine göre atanmıştır. Ayrıca veri setinde görüldüğü gibi RTT değerleri çok farklılık göstermektedir. Bunu engellemek için bu aralık K-means algoritmasından elde edilen merkez değerlerine göre aralıklarındırılmıştır.

Bir günün belli saat aralığında elde edilmiş RTT örnek değerleri için farklı sınıflandırma algoritmalarından elde edilen sonuçlar Çizelge 1’de verilmiştir. Çizelgede de görüldüğü gibi çoğu RTT değeri en az iki sınıflandırma algoritması tarafından aynı sınıfa dâhildir. En uygun senkronizasyon zamanına karar verirken sistemin etkinliği açısından bu bilgi göz önüne alınabilir.

Çizelge 1: RTT değerlerinin sınıflandırılması

RTT Değeri (msec)	K-Means	Bayes	C4.5
40	Orta	Orta	Düşük
16	Düşük	Düşük	Düşük
1	Düşük	Düşük	Düşük
45	Orta	Orta	Düşük
33	Orta	Düşük	Düşük
36	Orta	Düşük	Düşük
109	Yüksek	Yüksek	Yüksek
77	Orta	Orta	Yüksek
16	Düşük	Düşük	Düşük

40	Orta	Orta	Düşük
161	Yüksek	Yüksek	Yüksek
19	Düşük	Düşük	Düşük
2	Düşük	Düşük	Düşük
9	Düşük	Düşük	Düşük
48	Orta	Orta	Orta
18	Düşük	Düşük	Düşük
13	Düşük	Düşük	Düşük
12	Düşük	Düşük	Düşük
25	Düşük	Düşük	Düşük
41	Orta	Orta	Düşük
25	Düşük	Düşük	Düşük
56	Orta	Orta	Orta
11	Düşük	Düşük	Düşük
26	Düşük	Düşük	Düşük
62	Orta	Orta	Orta

Elde edilen sonuçların birbirleriyle tutarlılıkları Eşitlik (5) kullanılarak elde edilmiştir.

$$\frac{\text{Sınıflandırıcılardaki ortak RTT adedi} \times 100}{\text{Toplam Kayıt Adeti}} \quad (5)$$

İkili tutarlılıklar şu şekildedir:

Algoritmalar	Tutarlılık oranları
K-Means ve C4.5	%78
Naive Bayes ve C4.5	%83
K-Means ve Naive Bayes	%95
Hepsi	%72

6. Sonuç ve Tartışmalar

Bu çalışmada farklı yerlerde bulunan aynı verilerin tutarlılığı sağlanırken senkronize edilme zamanlarına karar verirken RTT değerleri ele alınmıştır. Bu değerler farklı sınıflandırma algoritmalarından geçirilerek içinde bulunulan an için sistemin senkronizasyon işlemine uygun olup olunmadığı kararı alınmıştır. Tek bir sınıflandırma algoritmasına göre karar vermek yerine aynı sınıfı gösteren en az iki yaklaşımı kullanmak sistemin etkinliği açısından daha önemlidir. Önerilen yaklaşım sadece senkronizasyon işlemlerine karar veren sistemler için değil, aynı zamanda bir düğümün haberleşme için uygun olup olmadığını araştıran her sistem için kullanılabilir. Önerilen sistem açık kaynaklar ile geliştirildiği için isteğe göre kolayca manipüle edilebilir.

Gelecek çalışmalarımızda bir senkronizasyon işleminden sonraki senkronizasyon işleminin ne zaman yapılmasına karar veren bir yöntem üzerinde çalışacağız. Sistemin yoğunluğuna göre bir sonraki senkronizasyon işlem zamanına karar vermek sistemi devamlı haberleşme için uygunluk testine tabi tutmaktan daha efektif bir sonuç üreteceğini düşünmekteyiz.

Teşekkür

Bu çalışma, TÜBİTAK tarafından 1003 Fatih Projesi Çağrısı kapsamında EEEAG 113E033 nolu proje ile desteklenmektedir. Desteginden dolayı TÜBİTAK'a teşekkür ederiz.

7. Kaynaklar

- [1] Kaya, F., Eken, S., İlhan, Z., Kavak, A., Sayar, A., Kocasarac, U., Şahin, S., "A Comparative Study of Signaling Protocols for Data Management and Synchronization in Fatih Project with School Level Cloud Proxy Server Deployment," *IEEE 3rd Symposium on Network Cloud Computing and Applications(NCCA'14)*, Roma, İtalya, s. 133-136, 2014.
- [2] Eken, S., Kaya, F., İlhan, Z., Sayar, A., Kavak, A., Kocasarac, U., Şahin, S., "Analyzing distributed file synchronization techniques for educational data", *10th IEEE International Conference on Electronics, Computer (ICECCO'13)*, Ankara, Turkey, s. 318-321, 2013.
- [3] Eken, S., Kaya, F., Sayar, A., Kavak, A., "Tracking a single node's availability for communication by means of observing local system resources," *International Symposium on INnovations in Intelligent SysTems and Applications (INISTA'14)*, Alberobello, İtalya, 2014.
- [4] <http://nmwg.internet2.edu/nmwg-tools.html> (Erişim Tarihi: 20 Haziran 2014).
- [5] Chi Ming, L., "Non-Intrusive Measurement in Packet Networks and its Applications," Ph.D. Theses, Department of Electronic Engineering, Queen Mary, University of London, 2004.
- [6] Andreozzi, S., Ciuffoletti A., Ghiselli, A., Antoniadis, D., Polychronakis, M., Markatos, E.P., Trimintzios, P., "On the Integration of Passive and Active Network Monitoring in Grid Systems," *Integrated Research in GRID Computing, CoreGRID Integration Workshop 2005 (Selected Papers)*, Pisa, Italy, s. 147-161, 2005.
- [7] Paxson, V., "End-to-End Internet Packet Dynamics," *IEEE/ACM Transaction on Networking*, vol. 7, s. 277-292, 1999.
- [8] Dovrolis, C., Ramanathan, P., Moore D., "What do Packet Dispersion Techniques Measure?," *20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2, s. 905-912, 2001.
- [9] Jacobson, V., "pathchar: A Tool to Infer Characteristics of Internet Paths," 1997.
- [10] Lai, K., Baker, M., "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," *ACM Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, s. 283-294, 2000.
- [11] Claffy, K.C., Dovrolis, C., "DOE SCiDAC Proposal: Bandwidth estimation: measurement methodologies and applications", Jul. 2001.
- [12] Oetiker, T., "MRTG: Multi Router Traffic Grapher," <http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html>.
- [13] Hasib, M., Schormans, J.A., "Limitations of Passive & Active Measurement Methods in Packet Networks," *London Communications Symposium (LCS)*, 2000.
- [14] <http://iperf.fr/> (Erişim Tarihi: 21 Haziran 2014).
- [15] <http://ftp.arl.army.mil/~mike/tcp.html> (Erişim Tarihi: 21 Haziran 2014).

- [16] <http://nmwg.internet2.edu/nmwg-tools.html> (Erişim Tarihi: 21 Haziran 2014).
- [17] <http://www.cs.bu.edu/~crovella/src/bprobe/Tools.html> (Erişim Tarihi: 21 Haziran 2014).
- [18] Jacobson, V., “Pathchar: A Tool to Infer Characteristics of Internet Paths,” <ftp://ftp.ee.lbl.gov/pathchar/>, (Erişim Tarihi: 22 Haziran 2014).
- [19] Dovrolis, C., Jain, M., “Pathload: a measurement tool for end-to-end available bandwidth,” *In Proceedings of Passive and Active Measurements (PAM) Workshop*, 2002.
- [20] Prasad, R., Dovrolis, C., Murray, M., Claffy, K., “Bandwidth estimation: metrics, measurement techniques, and tools,” *Network*, vol. 17, no. 6, s. 27-35, 2003.
- [21] Mitchell, T., *Machine Learning*, McGraw-Hill, 1997.
- [22] Alpaydın, E., “ZEKİ VERİ MADENCİLİĞİ: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri”, *Bilişim 2000 Eğitim Semineri*, 2000.
- [23] Dursun, Ö. O., *Meteorolojik Verilerin Veri Madenciliği ile Değerlendirilmesi*, Yüksek Lisans Tezi, Fırat Üniversitesi, ss. 10-11, 2005.
- [24] Quinlan, J.R., “Learning decision tree classifiers,” *ACM Computing Surveys (CSUR)*,” vol. 28, no. 1, s. 71-72, 1996.
- [25] Hunt, E.B., Marin, J., Stone, P.T., *Experiments in Induction*. Academic Press, 1966.
- [26] http://en.wikipedia.org/wiki/Round-trip_delay_time (Erişim Tarihi: 23 Haziran 2014).