T. C.
İSTANBUL TEKNİK ÜNİVERSİTESİ
Bilişim Enstitüsü
Bilgisayar Bilimleri Anabilim Dalı

BİLGİSAYAR MİMARİSİNDE YENİ YAKLAŞIMLAR

Prof. Dr. Bülent Örencik

"Emniyet- Kritik Uygulamalara Yönelik Bilgisayar Mimarileri"

Dönem Projesi Raporu

Karin Biricikoğlu
704061006

İÇİNDEKİLER

ŞEKİL LİSTESİ

1. Giriş

1.1 Kritik Sistemler

Çoğu yazılım kontrollü sistemin hatası kullanıcıyı zor durumda bırakır, fakat bu hatalar çok ciddi ve uzun süreli hasarlara yol açmaz, fakat bazı sistemlerin hatası ciddi ekonomik kayıplara, fiziksel hasarlara veya insan hayatını tehdit edebilecek sonuçlara neden olabilir.

Hatası neticesinde kayıpların çok büyük olduğu bu sistemlere kritik sistemler(critical systems) denir.

1.2 Kritik Sistemlerin Özellikleri

- Yüksek güvenilirlik
- Kullanılırlık(avaliability)
- Doğruluk(reliability)
- Hatasızlık(safety)
- Güvenlik(security)

1.3 Kritik Sistemlerin Çeşitleri

Emniyet -kritik sistemler(Safety-critical systems) : Hatası yaralanma, ölüm veya büyük çevresel hasarlara yol açabilen sistemlerdir. Örneğin tren rotaları ve saatlerini düzenleyen sistemler.

Amacı kritik olan sistemler(mission-critical systems) : Hatası sistemin amacını gerçekleştirememesine neden olan sistemler. Örneğin uzaya fırlatılan bir aracın rotasını belirleyen sistem.

İşi(kullanıcısı) kritik olan sistemler(business-critical systems): Hatası kullanan şirketin hata yapmasına yol açan sistemler. Örneğin bir bankanın müşteri hesap sistemi

1.4 Güvenilirliğin (Dependability) Genel Kavramları

Avaliability(Kullanılırlık): Herhangi bir zamanda sistemin çalışır durumda olması ve istenilen hizmetleri verebilmesi olasılığı.

Reliability(Doğruluk): Sistemin herhangi bir zaman dilimi içinde gerekli hizmetleri doğru bir biçimde verebilmesi olasılığı.

Safety (Hatasızlık): Sistemin insanlara veya çevresine zarar verme ölçüsü.

Security(güvenlik): Sistemin kasti ya da kazara yapılan etkilere(saldırı vb.) dayanabilme ölçüsü

2. DACAPO : Emniyet – Kritik Uygulamalar için Dağıtılmış Bilgisayar Mimarileri

2.1 Açıklama ve Mimarisi

DACAPO, Avrupa yol trafik programı PROMETHEUS çatısı altında bulunan Dahili Araç Yapısı(VIA) alt programı kapsamında yer alan otomobil üreticileriyle birlikte geliştirilen hata-toleranslı, sınıflandırılmış gerçek zamanlı bir bilgisayar sistemidir. VIA'nın amacı ucuz maliyetli ve esnek, otomobil tabanlı bilgisayar ağı geliştirmektir. Ucuz maliyet, elektrik ve elektronik elemanların yerleştirilmesinin sadeleştirilmesiyle elde edilirken esneklik, mevcut elektronik sistemlerinde uygulanmış işleme fonksiyonlarının kabiliyetiyle ilişkilidir.

Bir otomobil tabanlı bilgisayar ağında güvenlik ve zamanlama açısından kontrolü en zor olan fonksiyonlar, direksiyon, durma, süspansiyon ve güç-treni kontrolleri gibi sayabileceğimiz otomobil dinamiklerini gerçek zamanlı kontrol eden fonksiyonlardır. Bu fonksiyonların tamamı zaman-tetikli görev olarak yerine getirmek için uygundur. Ayrıca geleneksel sonuç odaklı işlem yürütmesi ve iletişimi kabul edilebilir derecede güvenlik sağlamaz,bu durumda ihtiyaç duyulan zamanlamayı garanti edemez. Dolayısıyla DACAPO'nun gerekli temeli olarak devirli operasyon ve iletişim uygun görülmüştür.

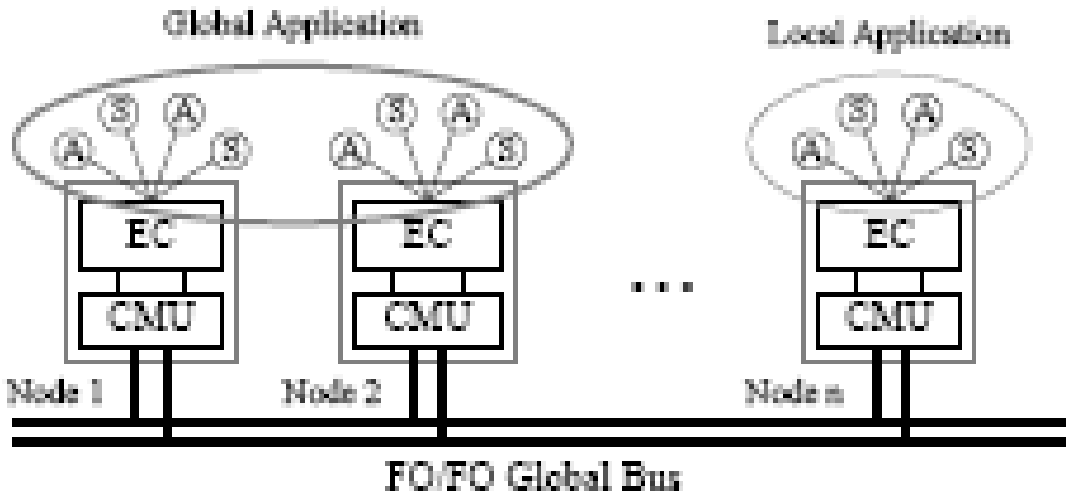
CAN, otomobil sistemlerinde geniş olarak kullanılan bir iletişim protokolüdür. Bu sistem, özellikle geçici hataların mevcudiyetinde, nispeten uzun kontrol gecikmelerinin bulunduğu sonuç odaklı anlaşma temelli karmaşık bir protokoldür. Buna rağmen CAN'i, zaman belirleyicili ağ yapısı içinde zemin projelerinin yerine getirilmesinde kullanmak mümkündür. CAN, sisteme gereksiz karmaşıklık ve maliyet getireceğinden DACAPO sistemlerinde kullanılmamıştır.

DACAPO sistemi, Şekil 1'de de görüldüğü gibi hata toleranslı ağlar üzerinde iletişim sağlayan küçük miktarlarda bilgisayar düğümlerinden oluşmuştur. Bu düğümler sensorlerin (S) ve harekete getiricilerin (A) toplandığı yerlere uzaysal olarak dağıtılmıştır. Mesela, her bir tekerleğin kenarında bir tane, motorun yakınında bir tane ve kontrol panelinin yanında bir tane olmak üzere dağıtılmış olabilir.

Her bir düğüm öncelikle bir gömülü kontrolcü, EC, ve bir iletişim ünitesi, CMU, içerir. EC ve CMU'nun her biri iki takım arıza-susturuculu fonksiyonel üniteden oluşur. Böylece her bir düğümü birkaç değişik yolla tekrar programlamak mümkündür. En kötü durumda, herhangi iki sonraki kalıcı arıza için, bir düğüm FO/FS(arıza-yürürlüklü/arıza-susturuculu) durumundadır. Aygıt, FO/FS ise birinci kalıcı arıza durumda tam olarak işlevsel ve ikinci arıza oluştuğunda ise sessizdir.
İletişim ağı iki seri bus içerir. Global bus sistemi, iki kalıcı arıza oluşmasına rağmen sistemin halen işlevsel olduğu anlamına gelen, FO/FO durumundadır.

Bir aracın kontrol fonksiyonlarında doğal bir fazlalık vardır; yerel bir kontrol fonksiyonu doğru şekilde çalışmasa bile, eğer arızalanan fonksiyon önceden belirlenebilir bir biçimde hareket ederse, aracı düşük bir performansla kontrol etmek mümkündür. Mesela aracı durdurmak, tek bir tekerleğinde durdurmak için bir kuvvet olmasa bile, mümkün olabilecektir. Ancak, yüksek derecede bir güvenlik sağlayabilmek için ilk kalıcı arızadan kısa bir süre sonra aracı tamir etmek gerekmektedir. Bu, aracın güvenliğini tehlikeye atacak ikinci bir arızanın oluşum olasılığını azaltacaktır. Güvenliği daha da arttırmak için sistem, düşük bir biçimde hareket edecektir.

Her bir düğüm, yerel sensorleri okuyan, çıkışı hesaplayan ve hareket ettiricileri kontrol eden devirli gerçek zamanlı güvenlik tenkitli kontrol uygulamalarını çalıştırır. Ancak durum-tetikleyicili uygulamalar da yürürlüğe konulabilir. Zaman tetikleyicili düğümler arasındaki iletişim ve kritik uygulama işlevleri, çalışma zamanından önce devirli olarak listelenir. Kritik olmayan ve durum-yürütücülü uygulama işlemleri çalışma zamanında listelenir ve ele alınır.



Şekil 1. DACAPO sisteminin mimarisi

2.2 Sistem Operasyonu

Bir düğümdeki işlem çalıştırıcının ve düğümler arasındaki iletişiminin operasyon ilkesi, çalışma zamanı öncesi listelemeye ve devirli operasyona bağlıdır. Buradaki birinci güdü, basit bir tasarım sonuçlandırmasının dışında, devirli önceden listelenmiş bir sistemin hedef uygulama için çok değerli faydalar sağlamasıdır.

- Zaman belirleyicilik, gerçek zamanlı sınırlara ulaşılabilirliği garantiler.
- Mesaj iletim arızasının etkileri normal olarak zamanla limitli olmasıdır.
- İletişim programının erken gelen bilgisi birçok muhtemel arızanın hızlı keşfedilmesini sağlar.

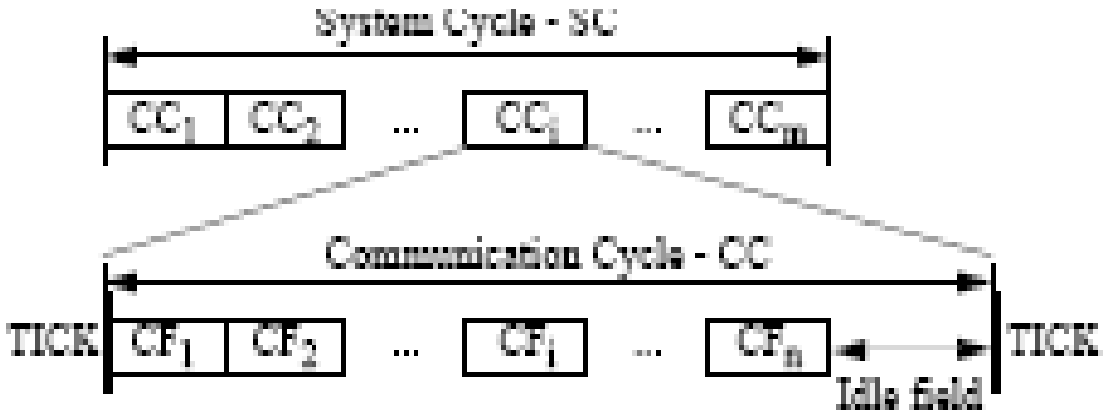
Bundan başka devirli operasyon, doğal olarak periyodik işlemler olan sensor örnekleme, hesaplamalar, harekete getirici kontrolleri gibi kontrol uygulamalarıyla gayet uyumludur.

2.2.1 İşlem Listeleme

Derleme zamanında her bir düğüm için çalıştırılabilir kod blokları oluşturulmaktadır. Bu kodun çalıştırılması, tüm uygulama işlemlerinin sıralı çalıştırılmalarına bağlıdır. Bir işlemin periyoduna göre bu kodun çalıştırılması bloğun çalıştırılması sırasında bir veya iki kez listelenmiş olabilir. Blok sürekli olarak çalıştırılır. Örneğin çalıştırma

bloğun sonuna ulaştığında, tekrar başa döner. Bloğun çalıştırılmasının bir diğer adı da sistem döngüsüdür (SC).

SC, bir veya fazla işlemin çalıştırılmasına izin veren ve iletişim döngüsü (CC) adı verilen “m” tane zaman bölmesine ayrılmıştır. CC’ deki işlem çalıştırmanın başlangıcı bir zaman sinyali ile tetiklenir, TICK. TICK her bir düğümdeki iletişim ünitesi tarafından oluşturulur. Her bir CC, adı iletişim çerçevesi (CF) adı verilen “n” tane zaman bölmesine bölünmüştür. Her bir iletişim çerçevesi sırasında her bir düğüme, o düğümün işlem listelemesine tekabül eden sabit iletişim listelemesine göre, global buslar üzerinden bilgi gönderme izni verilir. Şekil 2 ‘ de sistem ve iletişim döngüsü yapısı gösterilmiştir.

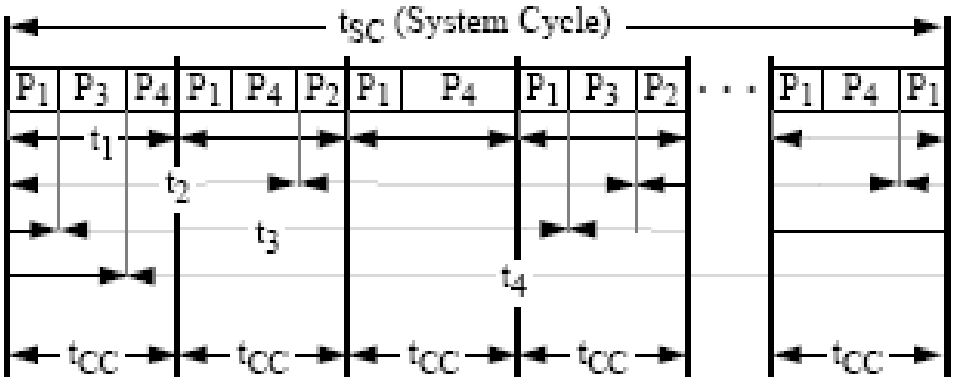


Şekil 2 . Sistem ve İletişim Döngüsü

Geliştirme yazılımının, sistem döngüsü sırasında değişik periyotlarda çalışan birkaç işlemin (P_1-P_4) gösterildiği (Şekil 3) benzeri kodlar oluşturabileceği farz edilmiştir. P_1’e en mümkün olan yüksek periyot değeri (periyot zamanı $t_1=t_{cc}$) verilmiştir, P_2 ve P_3’e sırasıyla $t_2=2- t_{cc}$ ve $t_3=3- t_{cc}$ periyot zamanları verilirken P4’e en düşük periyot (periyot zamanı $t_4=t_{sc}$) verilmiştir. Dikkat edilmelidir ki, eğer bir işlemin çalıştırma zamanı, bir iletişim döngüsündeki mevcut olan çalıştırma zamanını aşarsa, işlem bölünür ve birkaç iletişim döngüsünde çalıştırılır. Bu P_4 için geçerli olan durumdur.

Derleme zamanında, zaman-tetikli işlemler önce çalıştırılacakları bölümlere atanırlar. Kullanılmayan zaman bölümleri ise daha sonra durum-tetikli işlemlere atanır.

DACAPO ’daki devirli operasyonlar ve önceden listelenmiş kritik işlemler, gömülü kontrol sistemlerine yerleştirilen belirlemeleri karşılamaya kolay bir yolla olanak sağlar.



Şekil 3. Uygulama işleminin düğüm üzerinde çalışması

2.3 Düğüm Yapısı

DACAPO içindeki bilgisayar düğümleri temel olarak, birbirleriyle iki düğüm bus’ı (Şekil 1) tarafından bağlanan gömülü kontrolcü (EC) ve iletişim ünitesi (CMU), adındaki iki mantık bloğundan oluşur. Gömülü kontrolcü (EC) iki adet arıza-susturuculu I/O arabirimi içeren I/O ünitesi, iki arıza-susturuculu durum hafızası (SM) içeren hesaplama ünitesi (CU), iki arıza-susturuculu işleme elemanı (PE) ve bir FO/FS çapraz kaplama ünitesinden (CCU) oluşur (Şekil 4). İletişim birimi iki adet arıza-susturuculu global iletişim arabiriminden (GCI) oluşur.

Düğüm iki adet arıza-güvenli güç kaynağıyla (PS) beslenir. Şekil 4’te, taralı üniteler taralı güç kaynağıyla beslenirken geri kalanlar ise diğer güç kaynağıyla beslenmektedir.

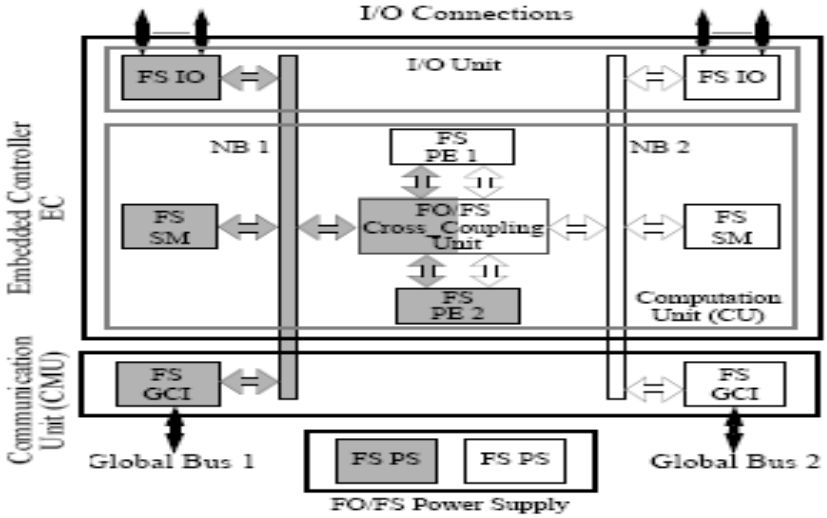
Bir DACAPO düğümü, FO/FS çapraz kaplama ünitesi tarafından birleştirilmiş iki arıza-susturuculu yarının birleşmesiyle oluşan FO/FS modül olarak düşünülebilir. İletişim ünitesinin görevi, düğümleri birbirine bağlamak ve global bus üzerinde gerekli bilgi transferlerini sağlamaktır. Bunun yanı sıra, durum bilgisi sağlar ve TICK sinyalini üretmekle görevlidir. Her iki GCI’dan biri global bus üzerindeki iletişimi sağlar. GCI, bir çift-uçlu RAM’den (DP-RAM), bir iletişim arabiriminden ve bir global clockdan (GC) oluşur (Şekil 5).

Her bir düğümün iki tane arıza-susturuculu I/O arabirimi vardır. Birlikte düğümü, sensorler ve hareket ettiricilerin bulunduğu yerel kümesine birleştirirler. I/O arabirimi

ya direkt arabirim ya da yerel ağ arabirimi olabilir. Her bir PE, tamamen senkronizasyon içinde çalışan ve yerel hafızaları bulunan iki mikroişlemciden oluşur. Bunların sonuçları, arızaların tespiti için sürekli karşılaştırılır. Eğer bir hata tespit edilirse, PE, düğüm yoluna (busına) doğru sessizleşir.

Hesaplama ünitesindeki durum hafızaları, bir iletişim döngüsünden diğerine kadar hatırlanması gereken durum bilgilerini saklamak için kullanılır.
CCU, PE'nin bus sırasının ve düğüm buslarının arasında yer alır. CCU, PE'yi herhangi bir düğüm busına bağlayabilir. Her bir iletişim döngüsünde ancak bir tek PE düğüm buslarını kontrol edebilir. CC'deki her bir uygulama işlemi için aktif olan PE, işlemi çalıştırmak için gerekli bilgiyi elde etmek için durum hafızasını okumaya başlar. Bir işlemin çalıştırılmasından sonra PE, bilgiyi tekrar durum hafızasına yazarak durur. Bunun sebebi her iki PE de işlemi çalıştırırken sırayla hareket eder, örneğin bir PE, diğer PE'nin bıraktığı yerden devam edebilmelidir.

Eğer her iki PE de arızasızsa, her bir iletişim döngüsünün başında değişerek uygulamanın yürütülmesinde ve düğüm buslarının kontrolünde sırayla görev alırlar. Bu değişim TICK sinyali tarafından tetiklenir. Bir PE çalışırken, diğeri beklemededir. Arızasız durumlarda, PE 1 öncelikli olarak birinci düğüm busına bağlı ünitelerden okur ve PE 2 öncelikli olarak ikinci düğüm busına bağlı ünitelerden okur. Ancak her bir PE, ulaşmak istediği NB'yi seçmekte özgürdür. Her PE, her zaman hem durum hafızasına hem de global iletişim arabirimine yazar. Karşılaştırıldığında bir PE'yi yedekte bekletme ve diğerini hatalı olana kadar çalıştırma, hesaplama ünitesinin her parçasının devamlı olarak denetlenmesine olanak sağlar. Eğer yedek meşgulse, o ünite aktif hale gelene kadar hata tespit edilemeyebilir. İlk hatanın gecikmeli tamiriyle birlikte ikinci ve tehlikeli hatanın oluşma riski artar. Ayrıca her iki durum hafızasına bilgi saklanması, bir kontrol yeri meydana getirilmesiyle çok benzer bir hareket olduğundan, değiştirme stratejisi arızaların iyileştirilmesini kolaylaştırır.



Şekil 4 . DACAPO Düğüm Yapısı

2.3.1 İyileştirme ve Arıza Yönetimi

Her bir düğüm, birçok donanım arıza tespit mekanizmasını içinde barındırır. Bu mekanizmalar, hem düğümün alt ünitelerindeki hem de düğümler arasındaki iletişim sırasındaki arızaları tespit eder. Bir arıza tespit edildiğinde, her PE bir arızanın oluştuğundan haberdar edilir. DACAPO 'nun periyodik doğasına uygun olarak, başarısız işlemler en son olarak bir sonraki SC' de yeniden çalıştırılır. Böylece eğer bir arıza ilk defa tespit edilmişse, aşağıdaki önlemler alınır;

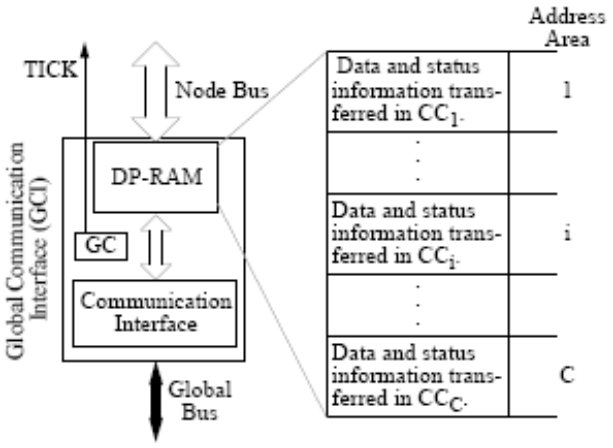
- Geçici bir hatanın arızaya sebebiyet verdiği farz edilir.
- Mevcut durumda iyileştirme yapılır (basit bir iyileştirme stratejisi kullanılarak).
- Eğer arıza kısa zaman aralıklarıyla tekrar ediliyorsa, hangi alt sistemlerin ya da elemanların hatalı olduğunu anlamak için bir tetkik yapılır. Arızalı ünite hatalı olarak etiketlenir ve eğer mümkünse sistemin dışında yeniden programlanır.

Hataları kontrol altına almak, arızanın sistem içindeki bir noktadaki merkezinden kullanıcıya verilen servise etki edebilecek noktaya ilerlemesini önleyen bir tasarım özelliğidir. Yazılımdaki I/O ayrımına, güvenlik duvarına, donanıma ve yazılıma uygulanan arıza-susturucu üniteler kullanılarak sistemin hem yazılım hem de donanım elemanları hataların kontrol altına alındığı bölgelere gruplanır.

2.4 İletişim Ünitesi

Daha öncede belirtilen iletişim ünitesi bağımsız olarak iki busda çalışan iki GCI'dan oluşur. Bir iletişim döngüsü sırasında her bir GCI ait olduğu busa bir mesaj gönderir. Busa ulaşım zaman katmanlıdır (Zaman Bölümlü Çoklu Ulaşım, TDMA). Mesajlar, düğüm kimlik numarasına göre sırayla gönderilir. Sistem döngüsü sırasında dağıtılan bilgi, DP-RAM içinde tutulur. Organizasyonu gönderim zamanlamasına göre tasarlanmıştır. Düğüm 0 tarafından ilk iletişim döngüsünde gönderilen bilgi, en son adreste gözükür (Şekil 5). Eğer belli bir parametre sistem döngüsü sırasında birden fazla kere gönderilmişse, değeri DP-RAM içinde birkaç konumda görülebilir.

Her mesaj, 100 bit uzunluğundaki bir yapıda 8 bytelık kullanıcı bilgisi taşır. Her bir yapı, mevcut döngü sayısını ve gönderen düğümün kimliğini içerir. Bu, görev yürütülmesinin ve iletişimin önceden listelenmiş olduğundan daha önce geçici olarak teması kaybetmiş diğer düğümlerin birbirleriyle hızlı bütünleşmesini sağlar. Bir 16 bitlik CRC kontrol, durum ve tasdik bitleri de içerir.



Şekil 5 . Global İletişim Arabirimi (GCI)

2.4.1 Zaman Temsili ve Senkronizasyonu

GCI küresel olarak kabul edilen bir zaman sağlayan global bir clock içerir. Mevcut zaman iletişim döngü sayısı ve düğüm numarası sayısından oluşur. İlgili sayaçlar, bus üzerindeki bit akışıyla eşzamanlı bit clock tarafından yürütülür. Her bir iletişim döngüsünün başlangıcında, iletişim ünitesi, gömülü kontrolcüye TICK sinyali sağlar.

İletişim bant aralığı cinsinden yüksek verim elde etmek için bilgi transferi bit-senkrondur ve mesajlar arasındaki boş zaman kısa tutulur. Bu farklı global iletişim arabirimleri arasında sıkı clock senkronizasyonu demektir. Düşük seviye senkronizasyonu ise Daisy-Chain senkronizasyon metodu kullanılarak elde edilir. Busa bir mesaj ulaştığında, tüm düğümler bit clocklarını sıfırlar. Tüm clocklar gelen mesajın başında senkronize edildiğinden, bilgi gövdesinin senkron kabulünü tehlikeye atmadan 10^{-2} oranında bir clock sürüklenmesine yaklaşılabilir. Daisy-chain senkronizasyonu, clock ayarlarının sadece tek bir clock okumasına göre yapıldığı bir geri çekim yapısına sahiptir. Hatalı clockların bulunduğu düğümlerin geri kalan yapının birleştirilmesini etkilememesi için çok fazla sapan clock okumaları atılır. Bu bir kabul penceresi kullanılarak yerine getirilir. Dar bir zaman aralığında (beklenen ulaşma zamanı etrafında temel alınan bir bitlik aralığa denk) ulaşmayan mesajlar atılır.

2.4.2 İyileştirme ve Arıza Yönetimi

İletişim ünitesi, iletişim hatalarını yöneten mekanizmalar içerir. Daha yüksek seviyelerde uzak düğümlerin durumunu izlemek için ayrıca birçok mekanizmalar da eklenmiştir. Uygulama seviyesinde koordineli düzeltme eylemlerinin gerçekleştirilebilmesi için hataların tüm düğümler tarafından tespit edilmesine ihtiyaç vardır.

Sistemin birleşmiş bir görünümünü elde etmek için ilk adım hangi global iletişim arabiriminin çalışır durumda olacağına karar vermektir. Arıza-susturuculu özellik yüzünden bu işlem arabirimlerin gönderilmesine eşittir. Alıcılar arasındaki geçici arızalara gidermek için iki başarılı GCI, iletilen mesajı tanır.
Böylece sadece tanıyan GCI dışındaki tüm GCI'lar orijinal mesajı almada başarısız olsa da, doğru bilgi gövdesinin gönderildiği konusunda bir uzlaşma sağlanır. Ne zaman ki ardı ardına birçok kez bir GCI mesaj gövdesini iletmede başarısız olursa, o GCI arızalı kabul edilir ve çalışır durumdaki diğer GCI'lerden ayrılır. Bu gecikme, geçici hatalar sonucunda oluşan mesaj karışıklığının, uzak GCI'lerin kalıcı başarısızlığı olarak algılanmasını engeller.

Bir gerçek zamanlı sistemde, önemli soru çeşitli kontrol uygulamaların çalışır durumda olup olmadığı sorusudur. Eğer mesela bir frenleme mekanizması bozulursa, frenlemeyle alakalı tüm düğümlerin bunu aynı zamanda tespit etmesi çok önemlidir. Bu amaç için bir takım durum bayrakları üzerinden GCI'lar arasında anlaşmayı garantileyen bir protokol desteği vardır. Gruptaki bir bayrak, her iletilen mesajla güncellenir. Böylece durum bayrakları, izlenen uygulamanın bilgi örnekleme oranına tekabül eden bir oranda tazelenir.

Mesaj kaybına önlemek için her bir bilgi elemanı bayat/güncellemeli bayrak taşır. Mesaj kaybı yüzünden yeni bilgi alınamazsa bu bayrak bayat olarak etiketlenir. Ayrıca eğer gönderen düğüm taze bilgiyi üretemiyorsa, DP-RAM'deki eski bilgi bayat bayrağı eşliğinde gönderilir. Bunun sonucu olarak o değişkeni kullanan uygulamalar DP-RAM'deki başka bir yerdeki örneği bulur ya da önceki bilgiye benzeyen yeni bir değer çıkarır.

Bir GCI' nın bir diğer GCI' dan mesaj alamamasının sebebi ya arıza sessizliğiyle sonuçlanan uzak ünitedeki bir arızaya ya da senkronizasyon kaybından ileri gelir. Eğer geri kalan düğümlerin yarısından azı doğru bir şekilde ulaştırılırsa, GCI sessizleşir ve çoğunluğa göre tekrar senkronize olur. Bu özellik senkron olmamış ortak GCI gruplarının operasyonlara devamını engeller. Bunu yerine tüm çalışır GCI'lar sonunda daha büyük bir grup olan senkron ünitelere dahil olur.

Senkronizasyona tekrar ulaşmak için bir GCI sessiz olduğu yeniden senkronizasyon safhasına girer ve açık kabul penceresine sahip diğer düğümlerden mesajı almaya çalışır böylece mesajların ulaşma zamanından bağımsız olarak kabul edilmesine olanak sağlar. Düğüm topluluğunun yarısından mesaj alındığında, GCI senkron üniteler grubunun bir parçası olur ve normal operasyonlara dönebilir.

Eğer yarıdan fazla düğüm topluluğu sırayla yeniden senkronize olmaya çalışırsa, tüm GCI' lar sonunda sessiz olur. Ne zaman bir düğüm, tüm bir iletişim döngüsü sırasında sessiz bir bus görürse, listelemede yer alan atanmış zaman bölümünden yerel zamanına uygun bir mesaj göndereceği iyileştirme safhasına girer. Kalan diğer düğüm topluluğu ilk mesaja göre senkronize olur ve iyileştirme safhasına girenler tekrar göndermeye devam eder. Düğüm topluluğunun yarısından alım yapılabildiği anda GCI normal operasyonlarına geri döner.

2.5 Güvenlik Önlemleri

Eğer sistemde hatalar varsa ve teşhis yazılımı tek bir hatanın daha sistemin feci bir şekilde bozulacağını öngörüyorsa, sistem düşük seviyeli operasyon durumuna girer.

Düşük durumda örneğin araç sadece sınırlı bir hızla kullanılabilir. Bu iki amaca hizmet eder, (i) sürücünün aracı tamir etmesi gerekir ve (ii) eğer ek bir hatadan dolayı sistem bozukluğu meydana gelirse, sonuçlar aracın yüksek hızlarda kullanımından daha az şiddetli olur. Düşük operasyon durumu ve tamirden önce kısa zaman birleşimi daha yüksek seviyede bir güvenlik sağlar. Sistemin güvenliğinin elde

edilmesi için, DACAPO sisteminin basitleştirilmiş Markov modeline uygun bir güvenilirlik analizi yapılır.

3. Zaman Tetikleyici Mimari (TTA)

TTA'nın temel amacı idare edilir bir gayretle inşa edilen güvenilir dağıtılmış uygulamaları mümkün kılmak için tüm düzgün düğümlere tutarlı dağıtılmış hesaplama temeli sağmaktadır.

Yüksek güvenilirlikteki uygulamaları olan dağıtılmış gömülü gerçek zamanlı sistemler alanında bilgi işleme için bir çerçeve oluşturur.

TTA düğümlerdeki (ECU) ve demetlerdeki büyük gömülü uygulamaları ayrıştırır ve her düğümde bilinen global zamanlı bir arıza tahammülü sağlar.
TTA tüm düzgün düğümlere tutarlı dağıtılmış hesaplama ve iletişimi sağlar. Ayrıca durum tutarlığını garantiler ve de hata tespiti gerçekleştirir.

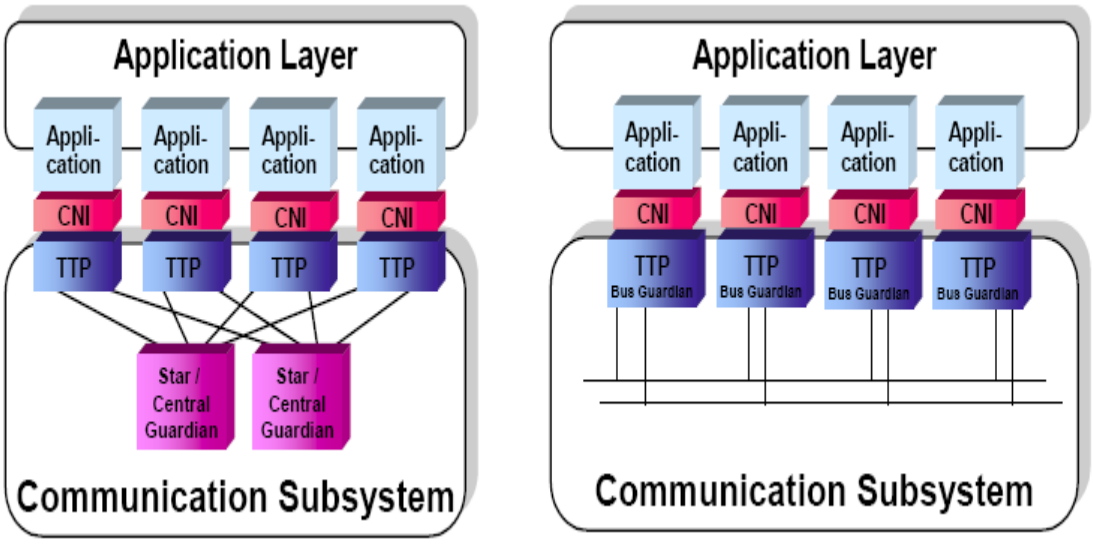
3.1 TTA ' nın Yapısı

TTA'nın bu en temel bloğu bir düğümdür. Bir düğüm kendi kendine yeten bir ünitenin içinde hafıza içeren bir mikro işlemciyi, bir giriş-çıkış alt sistemini, bir zaman-tetikleyicili iletişim kontrolcüyü, bir işletim sistemini ve ilgili uygulama yazılımını barındırır.

İki adet kopyalanmış iletişim kanalı, düğümleri bir küme oluşturmak için birbirine bağlar. Küme içindeki tüm düğümlere ait fiziksel bağlantı yapısı ve iletişim kontrolcüler, TTA' da otomatik olarak yer alan iletişim alt sistemini oluşturur ve apriori belirleyicili periyodik TDMA (Zaman Bölmeli Çoklu Ulaşım) listelemesini çalıştırır. İletişim alt sistemi apriori noktasında iletişim ağ arabiriminden (CNI) gelen durum bilgisi gövdesini okur (bu işleme çekme anı denir) ve bu bilgi gövdesinin önceki sürümünün üzerine yazarak apriori noktasındaki kümenin tüm alıcı düğümlerindeki CNI' lara gönderir (bu işleme servis anı denir). Periyodik olan çekme ve servis anları, bir kümedeki tüm iletişim kontrolörleri tarafından sürekli olarak bilinen MEDL (Mesaj Tanımlayıcı Liste) adındaki iletişim kontrolörü içinde yer alır.

TTA iki değişik ağ topolojisi üzerine yerleştirilmiştir: bus topolojisi (TTA-Bus) ve yıldız topolojisi (TTA-Star). TTA-Bus uygulamasına ait düğümler, saçma hataları önleyen yerel düğüm bus koruyucularıyla donatılmıştır. TTA-Star, bir kümenin tüm düğümleri tarafından paylaşılan akıllı merkezi koruyucuları uygular. Bu akıllı koruyucular, keyfi

düğüm arızalarını izole eder ve güvenlik-tenkitli uygulamaları destekler. Maliyet söz konusu olduğunda TTA-Star çok çekicidir çünkü her kanal için sadece bir bus koruyucu gereklidir. Bakınız Şekil 6.



Şekil 6 . TTA-Star ve TTA-Bus yapıları

3.2 Tasarım İlkeleri

Zaman-Tetikleyicili Mimarinin tasarımını yürüten ilkeler aşağıda listelenmiştir.

Tutarlı Dağıtılmış Hesaplama Platformu: Eğer bir düğüm diğer tüm düzgün düğümlerin aynı durumda çalıştığını varsayamazsa dağıtılmış algoritmaların tasarımı zahmetli olur çünkü karmakarışık anlaşma problemi uygulama seviyesinde çözülmelidir. Bu yüzden TTP direkt olarak iletişim alt sistemi seviyesinde tutarlı destek sağlar.

Arabirimlerin Birleştirilmesi – Geçici Güvenlik Duvarları: İyi bir mimari, birçok değişik durumlarda tekrar kullanılabilir küçük sayıdaki dikgen kavramlara zemin hazırlayarak karmaşık sistemlerin anlaşılması için gerekli olan gayretleri azaltmış olur. Zaman-tetikleyicili listelemesi tarafından yürütülen TTP, göndericinin iletişim ağı arabiriminden (CNI) alıcının CNI'larına bilgi gövdelerini taşırken otomatiktir. İtme-çekme örneğine göre gönderici bilgiyi kendi yerel CNI hafızasına koyabilirken alıcı bilgiyi kendi yerel CNI hafızasının dışından almalıdır. TTA'da hiçbir kontrol sinyali CNI'dan geçmediğinden (iletişim sistemi çekme ve servis anları için gerekli olan kontrol sinyallerini, global zamandan ve onun yerel MEDL tablosunda üretir), kontrol

hatası yayılması tasarım tarafından engellenmiş olur. Kontrol hatası yayılımını tasarımla önleyen arabirime geçici güvenlik duvarı denir.

Oluşturulabilirlik: Dağıtılmış gerçek zamanlı sistemlerde düğümler, yeni oluşan gerçek zamanlı sistemler sağlamak için iletişim sistemini kullanarak etkileşir. Bu yeni oluşan sistemler, düğümlerin arabirimlerinde gerçek zamanlı bilginin vakitli bir şekilde tedarik edilmesine bağlıdır. Bir mimari, eğer aşağıdaki dört ilkeyi sağlarsa geçici sahada oluşturulabilir olarak nitelendirilir:

1. Düğümlerin bağımsız geliştirilmesi
2. Yeni fonksiyonların bütünlenmesinden önceki servis istikrarı
3. Yeni oluşan servislerin üretilmesi için düğümlerin yapıcı bütünleşmesi
4. Kopya belirleme

Derecelendirebilirlik. TTA, karmaşık dağıtılmış gerçek zamanlı uygulamaların tasarımı için planlanmıştır. Birçok değişik fonksiyonları destekleyen bir karmaşık sistem, eğer belirli bir sistem fonksiyonunu anlamak için gerekli olan gayret sistemin büyüklüğünden bağımsız ise en etkili biçimde oluşturulabilir. Yatay tabakalandırma (soyutlama) ve dikey tabakalandırma (bölümleme), büyük sistemlerin karmaşıklığını çözmek için temeldir. TTA'da, CNI'lar fonksiyonları toparlar ve sadece işlevi için gerekli çevre özelliklerini görünür kılarlar.

3.3 Zaman-Tetikleyicili Protokol (TTP)

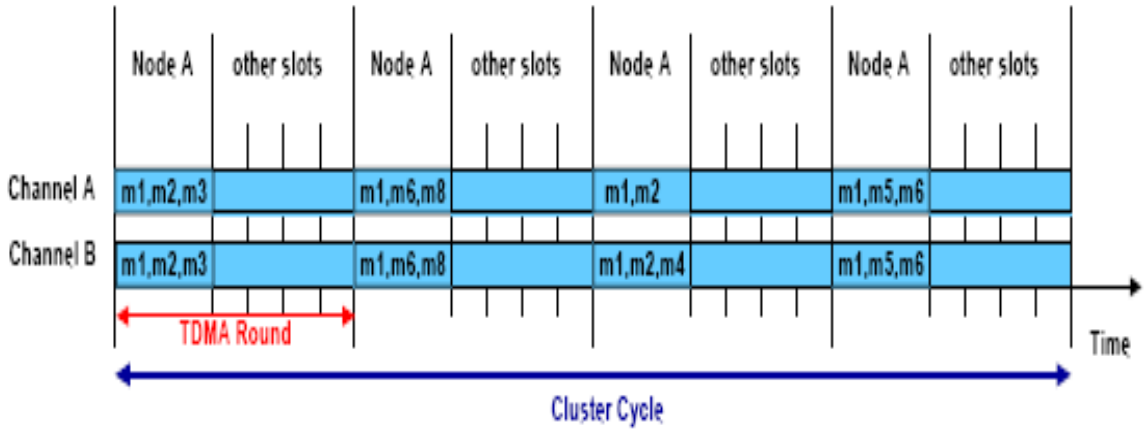
TTP, aşağıdaki servisleri sağlayan arıza-toleranslı zaman-tetikleyicili bir protokoldür:

1. Bilinen zamanlarda ve kümelerin düğümlerinin CNI'ları arasında en az sıkıntıyla özerk arıza-toleranslı mesaj iletimi, örneğin kopyalanmış iletişim kanalları üzerinden TDMA stratejisi görevlendirmesi kullanılarak bir ağdaki ECU'lar gibi.
2. Merkezi zaman serverına güvenmeden global zaman temeli oluşturan arıza-toleranslı clock senkronizasyonu.
3. Her doğru düğümü bilgi iletimi tutarlılığı hakkında bilgilendirmek için üye servisleri. Bu servis, iletişim sisteminde bir hata oluşmuşsa uygulamayı çabuk

bir şekilde bilgilendiren bir dağıtılmış onaylama servisi olarak görülebilir. Eğer durum tutarlılığı yok olursa, uygulama hemen haberdar edilir.

4. Protokol seviyesinde tahammül edilemeyen arıza hipotezi dışındaki hataların tespiti için grup sakınması.

TTP'de iletişim TDMA dizileri içine organize edilir. Bir TDMA dizisi bölümlere ayrılmıştır. İletişim sistemindeki her düğümün bir bölümü vardır - onun gönderme bölümü – ve her bir dizide bilgi gövdelerini göndermelidir. Genellikle birçok mesaj taşıyan bilgi gövdelerinin düğümlere tahsis edilmiş ebadı uzunluk bakımından 2 ile 240 byte arasında değişir. Küme döngüsü, TDMA dizilerinin tekrarlanan zinciridir; değişik dizilerde bilgi gövdelerine değişik mesajlar ulaştırılabilir ancak her küme döngüsünde tüm durum mesajı takımı tekrarlanır. Bilgi 24 bitlik bir CRC (Devirli Fazlalık Kontrolü) tarafından korunur. Çizelge, iletişim kontrolörünün içinde bulunan mesaj tanımlama listesinin (MEDL) içinde saklanır. Bakınız Şekil 7.



Şekil 7. Çerçeveler,bölmeler, mesajlar , TDMA serisi, küme yapısı

Clock senkronizasyonu, tüm düğümlere eşit bir zaman kavramı sağlamak için gereklidir. Bunu gerçekleştirerek gönderim listelemesindeki ortak bilgilerin kullanımı sağlanmış olur. Göndericinin ve alıcının clockları arasındaki farkı öğrenebilmek için her bir düğüm beklenen apriori ve doğru mesajın izlenen ulaşım zamanı arasındaki farkı ölçer. Bir arıza-tahammüllü ortalama bir algoritma bu bilgiye yerel clocka belirli aralıklarla düzeltme terimi hesaplayabilmek için ihtiyaç duyar. Böylece clock'un kümedeki diğer clocklarla eşzamanlı olmasını sağlanmış olur. Üyelik servis, bir arıza durumumda göndericinin çıkan bağlantısının mı yoksa alıcının gelen bağlantısının mı arızalandığını tespit edebilmek için dağıtılmış anlaşma algoritması kullanır.

3.3.1 Hata Toleransı

Eğer doğru bir şekilde programlanmış TTP tabanlı sistemlerin elemanları kontrol altına alınmış değişik bölgelerdeyse bu elemanlar keyfi olarak yerleştirilmiş olabilir. Bu varsayımlar altında iki rastlantısal bağımsız eleman arızası olma ihtimali, uygun bir asla vazgeçme (NGU) stratejisi tarafından idare edilebilecek nadir bir olay olarak nitelendirilir. Ancak unutulmamalıdır ki, iki ardışık tekli hataların keyfi olmamasını kesinleştirmek için çok kesin bir arıza tespit mekanizmasına ihtiyaç vardır.

Donanım hataları için ise TTP tekli düğüm hatalarını izole ve engel olmak için tasarlanmıştır. Bir bus koruyucu kullanılarak arızalı bir düğümün düzgün düğümlerin bilgi alışverişini engellemesi durdurulmuş olur. Bu bus koruyucu bir düğümün, bir TDMA serisi sırasında sadece bir kez gönderim yapabileceğini sağlar dolayısıyla iletişim araçlarını tekeline alan saçma hatalar probleminden kurtulmuş olunur. Ayrıca TTP, çoklu arıza senaryoları için bir NGU stratejisi yerine getirir; eğer bir düğüm arıza hipotezi tarafından tanınmayan bir arıza tepsi ederse, uygulamayı haberdar eder. Bundan sonra uygulama ya arızasız bir ortamda kapanır ya da dağıtılmış sistemin tüm düğümleri arasında kabul edilen bir tutarlı durum içeren bir ortamda yeniden başlar. Ayrıca bu mekanizma , TTP içinde iletişim alt sisteminde arıza toleransını ifade eder.

İletişim alt sisteminin bu mekanizmaları hatalı düğümlerin, düzgün düğümlerin iletişimini engelleyememesini temin eder ve uygulama için bir iletişim platformu olarak görev yapar. Uygulama seviyesinde arıza toleransı, arıza tolerans tabakası ve uygun bir uygulama tasarımı tarafından yerleştirilir. Arıza-toleransı, iki arıza-susturuculu düğüm üzerinde bir yazılım alt sisteminin kopyalanmasıyla gerçekleştirilebilir. Tekli bir rastgele düğüm arızası toleransı TMR (Üçlü Modüler Fazlalık) seçimleriyle kesinleştirilir.

3.3.2 Tutarlılık Desteği

Bilgi tutarlılığı, tasarımı ve karmaşık dağıtılmış sistemlerin geliştirilmesini büyük ölçüde kolaylaştırır. Tekli düğüm sistemlerinde tutarlılık sağlanmış gibi kabul edilir çünkü eğer düğüm düzgünse hafızaya yazılan bilgi tüm yazılım alt sistemlerine aynı zamanda hazır bulunur ve tüm alt sistemler aynı değeri okur. Dağıtılmış sistemlerde artık bu tür tutarlılığın varlığı savunulmaz. Bunun için iki sebep vardır: birinci olarak mevcut durumda etkisi olan mesaj iletim gecikmeleri dikkate alınmalıdır; mesajın tüm alıcılara zamanın aynı noktasında ulaşacağının garantisi yoktur. İkinci olarak, özel düğümler düşebilir veya mesaj kaybolabilir.

Yüksek miktardaki olası düğüm arızaları, iletişim arızaları veya zincirdeki ve iletim zamanındaki farklılıklar uygulama alt sisteminin mantığını büyük ve karmaşık yapar. Öyleyse durum tutarlılığı, aşağıdaki özellikleri destekleyen iletişim alt sisteminin temel bir servisi olarak desteklenmelidir. Hata toleransı geçerli olması şartıyla bir sistem süre tutarlıdır eğer

1. Tüm doğru düğümler aynı bilgi üzerinde hemfikirse.
2. Tüm düğümler doğru gönderici tarafından gönderilen bilgi üzerinde hemfikirse.
3. Tüm doğru alt sistemler alınan değeri aynı zaman noktasında iletirse.

Bir düğümün bilgiyi alıcı düğümler serisine gönderdiği varsayılır. TTA donanımdaki iletişim tutarlılığını direkt olarak protokol seviyesinde desteklemek üzere tasarlanmıştır. Aşağıda tutarlılığı destekleyen mekanizmalar anlatılmıştır.

3.3.3 Üyelik ve Tanıma

TTP tasarımındaki ana felsefe, protokolün bilgiyi devamlı olarak dağıtılmış sistemin tüm doğru düğümlerine göndermesi ve bir arıza durumunda iletişim sisteminin kendi kendine hangi düğümün arızalı olduğuna karar vermesidir. Bu özellikler üyelik protokolü ve tanıma mekanizması tarafında sağlanır.

TTP tabanlı kümenin her bir düğümü, doğru olduğu değerlendirilen tüm düğümler içeren bir üyelik listesi tutar. Bu bilgi yerel olarak başarılı (ya da başarısız) bilgi transferleri tarafından doğrulanır böylece alıcı düğümün diğer tüm düğümler üzerindeki yerel görüntüsünü yansıtır. Her bir transferle alıcı, CRC hesaplamalarında saklı olan ya da göndericinin transferinde bulunan gönderici üyeliğini görür ve kontrol eder.

Tanıma: İletimden sonra, A düğümü, iletinin alıcı tarafından (iletişim seviyesinde) kabul edilip edilmediğini anlamak için diğer düğümlerin onayını arar. Bu, ilk (ve muhtemelen ikinci) başarılı göndericinin üyelik listesinin kontrolüyle sağlanır. Eğer bu düğümler A düğümünü üyelik listesinde gösterirlerse, A'nın iletisinin başarıyla alındığını belirtir. Yoksa A'ya iletimin başarısız olduğu bildirilir. Zaman-tetikleyicili prensip sebebiyle durum mesajının tekrar gönderimi bir sonraki döngüde gerçekleştirilir.

3.3.3.1 Üyelik Tutarlılığı Kontrolü: TDMA dizisinin kesin round-robin tasarısı sebebiyle her bir düğüm bir TDMA dizisinde diğer düğümlerin üyelik listelerini görür ve kontrol eder. Değişik üyelik listesine sahip her gönderici yanlış olarak değerlendirilir. Böylece üyelikte birbirini kabul eden her bir düğümün tutarlılığını garantiler. Örneğin birbirleriyle başarılı bir şekilde iletişim sağlarlar.

3.3.3.2 Klik sakınması: Çoklu eleman hatalarını ve tutarsızlıkları tespit etmek ve asla-vazgeçme stratejisini desteklemek için klik sakınması mekanizması hazır bulunur. Bir düğümün her bir gönderiminden önce algoritma düğümün temel kliğin üyesi olup olmadığını kontrol eder. Düğümün azınlık kliğinden olması durumunda ise tutarsızlığa yol açan arıza-hipotezi dışında nadir bir arıza senaryosu meydana geldiği anlamı çıkar. Bu vaziyet arıza-durdur ya da arıza-çalışır aktivitelerini yürütmeye karar verebilecek uygulama yazılımına duyurulur.

Clock senkronizasyonuyla tesis edilen ortak zaman temelinin bu algoritmalarla birleşimi iletişim tutarlılığını sağlar. Bu durum zamanın aynı noktasında tüm doğru düğümlerin aynı bilgiyi aldıklarını garantiler. Dolayısıyla TTA uygulama yazılımına karmaşık dağıtılmış yazılım sistemlerinde etkili idare sağlayan çok güçlü bir programlama modeli sağlamış olur.

4. Sonuçlar

DACAPO adındaki dağıtılmış gerçek zamanlı bilgisayar sistemi, önceden listelenmiş işlem yürütme ve iletişimle birlikte devirli operasyon gibi özellikleri bu sistemi emniyet kritik gerçek zamanlı uygulamalar için uygun bir mimari yapar. Ayrıca sistem içine yerleştirilen Hata mekanizması ve kolay fakat etkili iletişim protokolü DACAPO'da elde edilen yüksek güvenilirlik ve emniyete katkı sağlamıştır.

TTA mimarisi ve TTP protokolü adres karmaşıklığı ve türetilebilirliği gibi yeni fonksiyonlara bir çözüm sağlamaktadır. TTP, TDMA stratejisini temel alır. Ayrıca, TTP' nin hata toleransı, tutarlılık desteği, üyelik ve tanıma, klik sakınması gibi özellikleri bu sistemi emniyet kritik uygulamalara yönelik bir mimari yapar.

Ayrıca clock senkronizasyonu ile zamanın aynı noktasında tüm doğru düğümlerin aynı bilgiyi aldıkları garantilenir. Dolayısıyla, TTA uygulama yazılımına, karmaşık dağıtılmış yazılım sistemlerinde etkili idare sağlayan çok güçlü bir programlama modeli sağlamış olur.

Referanslar

1. Time-Triggered Technology TTTech (www.tttech.com)

2. DACAPO: A Distributed Computer Architecture for Safety-Critical Control Applications
B. Rostamzadeh, H. Lönn, R. Snedsbøl, J. Torin