Sayfa 1

Celer Network: İnternet Ölçeğini Her Blockchain'e Taşıyın
ScaleSphere Foundation Ltd. ("Foundation")
15 Haziran 2018
Taslak, değişebilir.

Öz. Tıpkı 90'larda 56Kbps çevirmeli İnternet'in muhtemelen destekleyemediği gibi
4K video akışı, günümüzün blok zincirinin yetersiz ölçeklenebilirliği anahtar faktördür
kullanım durumlarını sınırlandırıyor. Mevcut blok zincirleri, her işlemin
Zincir üzerinde fikir birliğine ulaşmak için düğümlerin büyük çoğunluğu tarafından işlenmesi gerekir.
tam olarak "süper yavaş bir dağıtım sisteminin nasıl kurulacağı" dır. İronik olarak, zincir üzerindeki
fikir birliği şeması, herhangi bir düğüm tüm işlemi görebildiğinden, zayıf mahremiyete de yol açar
birbirlerinin tarihi. Yeni fikir birliği algoritmaları önerilmeye devam ederken
geliştirildiğinde, zincir üzerinde fikir birliğini temel sınırlamalarından kurtarmak zordur.
Zincir dışı ölçeklendirme teknikleri, karşılıklı güvensiz tarafların bir sözleşme yürütmesine izin verir
küresel blok zinciri yerine yerel olarak kendi aralarında. Katılan taraflar
işlem, çoklu imzalı sahtekarlığa dayanıklı zincir dışı çoğaltılmış durum makinesini korur,
ve yalnızca kesinlikle gerekli olduğunda zincir üzerinde fikir birliğine başvurur (ör.
bağlar bir eyalette uyuşmuyor). Zincir dışı ölçeklendirme, tamamen ölçeklendirmeyi desteklemenin
tek yoludur
Merkezi olmayan uygulamalar ("dApps") daha iyi gizlilik sağlayan ve hiçbir ödün vermeyen
güven ve ademi merkeziyetçilik garantileri. Blockchain kütlesi için dönüm noktasıdır
ve tüm ölçeklenebilir dApp'lerin arkasındaki motor olacaktır.
Celer Network, İnternet ölçeğinde, güvensiz ve gizliliği koruyan bir platformdur.
herkes hızla ölçeklenebilir dApp'ler oluşturabilir, çalıştırabilir ve kullanabilir. Bu bir standart değil
dalone blockchain, ancak mevcut ve geleceğin üzerinde çalışan ağ bağlantılı bir sistem
blok zincirleri. Innova ile benzeri görülmemiş bir performans ve esneklik sağlar.
zincir dışı ölçeklendirme teknikleri ve teşvikle uyumlu kriptoekonomi.
Celer Network, temiz soyutlamalara sahip katmanlı bir mimariyi kucaklar.
genelleştirilmiş bir durum kanalı da dahil olmak üzere her bir bileşenin hızlı gelişimi
ve hızlı ve genel zincir dışı durum geçişlerini destekleyen yan zincir paketi; kanıtlanabilir bir
daha yüksek bir büyüklük sırasına ulaşan optimum değer aktarımı yönlendirme mekanizması
son teknoloji çözümlerle karşılaştırıldığında iş hacmi; güçlü bir geliştirme çerçevesi
ve zincir dışı uygulamalar için çalışma zamanı; ve sağlayan yeni bir kriptoekonomik model
ağ etkisi, istikrarlı likidite ve zincir dışı ekosistem için yüksek kullanılabilirlik.

Sayfa 2

ÖNEMLİ: ÖNCE AŞAĞIDAKİ SORUMLULUK REDDİNİ TAMAMEN OKUMALISINIZ DEVAM EDİYOR

CELR'nin (" **Token** ") satışı (" **Token Satışı** ") , Celer ile ilişkili kriyotografik jeton
Bu teknik incelemede (" **Teknik Rapor** ") ayrıntıları verilen ağ , yalnızca aşağıdakiler için
amaçlanmıştır, yapılmıştır veya yönlendirilmiştir:
sadece belirli kişiler. Bu Whitepaper herhangi bir türden bir prospektüs veya teklif dokümanı değildir
ve
herhangi bir biçimde menkul kıymet teklifini, bir ticari tröstteki birimleri, bir kollektifteki birimleri
oluşturmayı amaçlayan
yatırım planı veya başka herhangi bir yatırım şekli veya herhangi bir yatırım şekli için talep
herhangi bir yargı alanı. Hiçbir düzenleyici makam, belirtilen bilgilerin hiçbirini incelememiş veya
onaylamamıştır.
bu Whitepaper'da. Bu Teknik İnceleme, herhangi bir düzenleyici otoriteye kaydedilmemiştir.
yargı yetkisi.
Bu Beyaz Bültende veya bunun bir bölümünde yer alan herhangi bir bilgiye erişerek ve / veya bu
bilgilere sahip olmayı kabul ederek
(duruma göre), cSpeed Ltd'yi (BVI Business Company Number
1976167) (" **Token Satıcısı** "):

(a)
Hariç Tutulan Kişi değilsiniz (aşağıda tanımlandığı gibi);
(b)
Token Satışının yasak olduğu, kısıtlandığı veya kısıtlandığı bir yargı bölgesinde bulunmuyorsunuz veya
tamamen veya kısmen kanunları, düzenleyici hükümleri uyarınca herhangi bir biçimde veya şekilde yetkisiz
gereksinimler veya kurallar;
(c)
bu Beyaz Bültenin tamamını okudunuz ve sizin için gerekli olan riskleri anladınız.
Tokenlerin satın alınması;
(d)
burada açıklanan sınırlamalara ve kısıtlamalara bağlı kalmayı kabul edersiniz; ve
(e)
Bu Beyaz Bültenin size yardımcı olmak amacıyla size teslim edilmek üzere hazırlandığını kabul edersiniz.
Jetonları satın alıp almayacağınıza karar verirsiniz ve tamamen referans içindir
sizinle Token Satıcısı arasında yasal ilişki kurma niyeti olmadan veya
Token Satıcısına karşı sizin tarafınızdan yasal olarak bağlayıcı veya uygulanabilir.
Herhangi bir kabul etmiyorsanız, lütfen bu Beyaz Bültenin içeriğine daha fazla devam etmeyin.
yukarıda. Lütfen "Önemli Uyarı" başlıklı bölüme ve "Sorumluluk Reddi Beyanı" başlıklı bölümlere bakın.
Sorumluluk "," Beyan ve Garanti Olmaması "," Sizin Tarafınızdan Beyan ve Taahhütler ",
"İleriye Dönük Beyanlar Hakkında Uyarı Notu", "Üçüncü Taraf Bilgileri ve
Diğer Kişiler "," Kullanılan Terimler "," Tavsiye Yok "," Daha Fazla Bilgi veya Güncelleme Yok "," Dağıtım ve Yayımlama "," Yatırım veya Kayıt Olmaması "ve" Riskler Ve
Bu Beyaz Bültende daha fazla ilerlemeden önce "belirsizlikler" dikkatlice.

---

3. Sayfa

İçindekiler

---

4. sayfa

5.Sayfa

1. Giriş

Pek çok modern ekonomik faaliyet, temelde bilgi akışı ve alışverişidir.
ve değer. Son iki yüzyılda, bilgi aktarımı,
güvercin ağları yoluyla ışık hızındaki sürekli akışlara kadar farklı olaylar
İnternet. Bununla birlikte, değer aktarımı kısmı ışık hızından uzaktır ve hala çok
ayrılmış finansal silolar tarafından kontrol edilen çok farklı olaylar. Bu uyumsuzluk bir
ekonomik evrimde yıkıcı darboğaz: bilgi ne kadar hızlı akarsa,
pahalı ve yavaş değerli işlem,
iki.
Güvensiz partiler arasında esasen devrimci bir güven soyutlaması,
Teşvik odaklı dağıtılmış bir fikir birliği içinde sonuçlanan blockchain teknolojisi,
Ayrılmış finansal siloları sökmek ve kapsamı önemli ölçüde genişletmek için temel
ve küresel değer akışının özgürlüğü. Ancak pratikte blockchain daha da sapıyor
tra ile karşılaştırıldığında düşük işlem gücü nedeniyle ışık hızı görüşünden uzak
ikili değer aktarım araçları. Ölçeklenebilirlik, engelleyen temel bir zorluktur
blockchain teknolojisinin toplu olarak benimsenmesi.

İnsanların, bilgisayarların ve mobil cihazların bulunduğu merkezi olmayan ekosistemlerle bir gelecek öngörüyoruz.
safra ve Nesnelerin İnterneti ("IOT") cihazları güvenli, özel ve güvensiz performans gösterebilir
büyük ölçekte bilgi-değer değişimi. Bunu başarmak için blok zincirleri
İnternetin ölçeğine uygun ve yüz milyonlarca veya milyarlarca trans-
saniye başına eylem. Bununla birlikte, mevcut blok zincirlerinin işlem hızı göz önüne alındığında (yani,
saniyede birkaç veya onlarca işlem),
İnternet blok zincirleri için mi? Cevap evet, ancak yalnızca zincir dışı ölçeklendirme ile.
Zincir üzerinde fikir birliği blockchain teknolojisinin temelini oluştururken, sınırlamaları
ayrıca açıktır. Bir anlamda, fikir birliği ölçeklenebilirliğin tam tersidir. Dağıtılmış herhangi bir
sistem, tüm düğümlerin her işlemde fikir birliğine varması gerekiyorsa,

mance daha iyi olmayacak (aslında, iletişim ek yükü nedeniyle çok daha kötü)
Her işlemi işleyen tek bir düğüme sahip merkezi sistem, yani
Sistem, en sonunda en yavaş düğümün işlem gücü nedeniyle darboğaz altındadır.
Zincir üzerinde fikir birliğinin ayrıca gizlilik üzerinde ciddi etkileri vardır, çünkü tüm işlemler
kalıcı olarak halka açıktır. Birkaç zincir üzerinde fikir birliği iyileştirmesi önerildi
parçalama ve çeşitli Proof-of-X mekanizmaları dahil. Blok zincirini yeniden
performans, ademi merkeziyetçilik, güvenlik ve

Sayfa 6

kesinlik, ancak zincir üzerinde fikir birliğinin temel sınırlamalarını değiştiremez.
İnternet ölçeğindeki blockchain sistemlerini daha iyi gizlilikle ve ödün vermeden etkinleştirmek için
güven veya ademi merkeziyetçilik konusunda, zincir içi fikir birliği iyileştirmelerinin ötesine
bakmalıyız.
Ölçeklenebilir dağıtılmış bir sistem tasarlamanın temel ilkesi,
farklı düğümler çoğunlukla bağımsızdır. Bu basit içgörü, bunu yapmanın tek yolunun
Merkezi olmayan uygulamaları tamamen ölçeklendirmek, işlemlerin çoğunu zincir dışına çıkarmaktır,
Zincir üzerinde fikir birliğinden mümkün olduğunca kaçının ve son çare olarak kullanın. İlgili
teknoloji
nikler arasında durum kanalı, yan zincir ve zincir dışı bilgi işlem Oracle bulunur. Ona rağmen
yüksek potansiyeller, zincir dışı ölçeklendirme teknolojisi, birçok teknik
ve ekonomik zorluklar çözülmeden kaldı.
Prime-time kullanımı için ölçeklendirme zincirli kapalı etkinleştirmek için, Celer Ağı teklif 1 , bir
coher-
İnternet ölçeğini mevcut ve gelecekteki blok zincirlerine getiren ent mimarisi. Celer
Ağ, yüksek performans sağlayan, dikkatle tasarlanmış zincir dışı bir teknoloji yığınından oluşur.
güçlü güvenlik ve gizlilik garantileri ve bir oyun ile ölçeklenebilirlik ve esneklik
herhangi bir yeni ödünleşmeyi dengeleyen teorik kripto ekonomik model.
1.1. Celer Teknoloji Yığını
Mevcut veya geleceğin üzerine inşa edilebilecek kapsamlı bir tam yığın platform olarak
Celer Network, blok zincirlerini birbirinden ayıran temiz katmanlı bir mimariyi kapsar.
hiyerarşik modüllere gelişmiş zincir dışı platform. Bu mimari büyük ölçüde
sistem tasarımını, geliştirmesini ve bakımını basitleştirir, böylece her bireyin
bileşen kolayca gelişebilir ve değişikliklere uyum sağlayabilir.
İyi tasarlanmış, katmanlı bir mimarinin, etkinleştiren ve sağlayan açık arabirimlere sahip olması
gerekir.
Aynı şeyi destekledikleri sürece her katmanda farklı uygulamayı teşvik edin
katmanlar arası arayüzler. Her katmanın yalnızca kendi işlevselliğini elde etmeye odaklanması gerekir.
İnternetin başarılı katmanlı tasarımından esinlenen Celer Network,
cStack adlı farklı blok zincirleri üzerine inşa edilebilen zincir teknolojisi yığını,
Aşağıdan yukarıya sırayla aşağıdaki katmanlardan oluşan:
• cChannel: genelleştirilmiş durum kanalı ve yan zincir paketi.
• cRoute: kanıtlanabilir optimum değer aktarım yönlendirmesi.
• cOS: zincir dışı etkin uygulamalar için geliştirme çerçevesi ve çalışma zamanı
1 celer: hızlı, Latince'de hızlı, ışık hızı için c

---

7. Sayfa

Şekil 1. Celer Network katmanlı mimari.
Celer mimarisi tüm katmanlarda yenilikçi çözümler sunar. Aşağıda vurguluyoruz
cChannel, cRoute ve cOS'un teknik zorlukları ve özellikleri.
cChannel. Bu, farklı kullanıcılarla etkileşime giren Celer Network'ün alt katmanıdır.
temeldeki blok zincirleri ve üst katmana ortak bir yukarı
güncel durumlar ve sınırlı zaman kesinliği. cChannel, durum kanalını ve yan zinciri kullanır

zincir dışı ölçeklendirme platformlarının temel taşları olan teknikler.
Bir durum kanalı, karşılıklı güvensiz tarafların zincir dışı bir program yürütmesine izin verir
ve güvenlik ve kesinlik garantileriyle en son üzerinde anlaşmaya varılan eyaletlere hızla yerleşin
zincir içi tahvil sözleşmeleriyle sağlanır. Başlangıçta Lightning Network tarafından tanıtıldı [9]
yüksek verimli zincir dışı Bitcoin mikro ödemelerini desteklemek için. Girişten beri
Lightning Network'te, farklı konuları ele alan birkaç araştırma çalışması yapılmıştır.
yönlendirme gibi ödeme kanalı ağları bağlamındaki sorunlar [10, 13, 16],
zaman kilidi optimizasyonu [5] ve gizlilik [6]. Bununla birlikte, zincir dışı ağ hala
erken aşama, modülerlik, esneklik ve maliyet açısından birkaç büyük zorlukla karşı karşıya
verimlilik. cChannel, bir dizi yeni özellik sunarak mevcut zorlukların üstesinden gelir.
• Genel zincir dışı durum geçişi. Zincir dışı işlemler keyfi olabilir
bağımlılık DAG ile durum geçişleri. Bu, Celer Network'ün
oyun, çevrimiçi açık artırma, in- gibi karmaşık yüksek performanslı zincir dışı dApp'ler
surance, tahmin piyasası ve merkezi olmayan borsalar.
• Esnek ve verimli değer aktarımı. Çoklu durum kanalı ve yan zincir
verimlilik ve kesinlik konusunda farklı ödünleşimlere sahip yapılar,
genel koşul bağımlılığı ile hızlı değer aktarımını destekler, minimum zincir içi
etkileşimler ve minimum fon kilitlenmesi.

---

8. Sayfa

• Saf zincir dışı sözleşme. Doğrudan ilişkili olmayan herhangi bir sözleşme
Zincir üzerinde birikintiler, zincir üzerinde herhangi bir işleme veya başlatmaya ihtiyaç duymaz.
bir anlaşmazlık tetiklendi. Her saf zincir dışı sözleşme veya nesnenin benzersiz bir
tanımlanabilir zincir dışı adres ve yalnızca aşağıdaki durumlarda blok zincirlerinde konuşlandırılması gerekir
yerleşik zincir dışı adres tarafından atanan bir zincir üstü adres ile gereklidir
çevirmen.
cRoute. Celer Network, yüksek düzeyde ölçeklenebilir dApp'ler için tasarlanmış bir platformdur ve
yüksek verimli değer aktarımı. Zincir dışı değer transferi temel bir gerekliliktir
birçok zincir dışı uygulamada. Celer Network dApp'leri destekleyebilirken
ödeme çözümlerinin ötesinde, zincir dışı uygulamalar için de çığır açan iyileştirmeler yapar.
ödeme yönlendirmesi, değerin ne kadar ve ne kadar hızlı aktarılabileceğini doğrudan belirlediği için
ekosistem içinde yer alıyor.
Mevcut tüm zincir dışı ödeme yönlendirme teklifleri [3, 4, 10, 12, 13, 16] kaynar
kötü sonuçlara ulaşabilecek geleneksel "en kısa yol yönlendirme" algoritmalarına kadar
zincir dışı bir ödeme ağındaki temel farklılıklar nedeniyle performans
bağlantı modeli. Bir bilgisayar ağının bağlantı kapasitesi durumsuz ve kararlıdır (ör.
geçmiş yayınlardan etkilenmez). Bununla birlikte, zincir dışı bir ödemenin bağlantı kapasitesi
ağ durum bilgilidir (yani, zincirdeki mevduatlar ve geçmiş ödemeler tarafından belirlenir),
topolojinin ve bağlantı durumlarının sürekli olduğu son derece dinamik bir ağa yol açar
değiştirme. Bu, geleneksel en kısa yol yönlendirme algoritmalarının yakınsamasını zorlaştırır,
ve bu nedenle düşük verim, uzun gecikme ve hatta kesintiler sağlar.
Bu temel zorluğun üstesinden gelmek için, Celer Network'ün ödeme yönlendirme modülü,
cRoute, ödemeyi yönlendiren Dağıtılmış Dengeli Yönlendirme'yi (DBR) sunar
dağıtılmış tıkanıklık gradyanlarını kullanan trafik. Birkaç benzersiz özelliği vurguluyoruz
DBR (ayrıntılar § 3.2.2'de).
• Muhtemelen optimum verim. Herhangi bir küresel ödeme gelişi için bunu kanıtlıyoruz
oranı destekleyebilen bir yönlendirme algoritması varsa, DBR
bunu başar. Değerlendirmemiz, DBR'nin 15 kat daha yüksek verim sağladığını gösteriyor
ve son teknoloji çözümlere kıyasla 20 kat daha yüksek kanal kullanım oranı 2 .
• Şeffaf kanal dengeleme. "Kanalları dengede tutmak"
Lightning Network'ün teklifinden bu yana hedefimiz. Ancak, mevcut girişimler
2 Her bir zaman diliminde aktarılan fon miktarı ile tüm kanalların toplam yatırılan miktarı arasındaki oran.

Sayfa 9

kanal dengeleme, yoğun zincir içi veya zincir dışı gerektiren buluşsal yöntemler içerir
zayıf garantilerle koordinasyon. DBR, kanal dengeleme sürecini yerleştirir
yönlendirme ile birlikte ve herhangi bir ihtiyaç duymadan ağı sürekli olarak dengeler
ek koordinasyon.
• Tamamen merkezi olmayan. DBR algoritması tamamen merkezi olmayan bir algoritmadır
her bir düğümün, eyalet kanalı ağındaki komşularıyla yalnızca "konuşması" gereken yerlerde
topoloji. DBR protokolü ayrıca mesajlaşma maliyetini düşürür.
• Başarısızlık direnci. DBR algoritması, arızalara karşı oldukça sağlamdır:
maksimum desteği destekleyerek yanıt vermeyen düğümleri hızlı bir şekilde algılayabilir ve
uyarlayabilir
kalan kullanılabilir düğümler üzerinden olası verim.
• Mahremiyetin korunması. DBR algoritması ile sorunsuz bir şekilde entegre edilebilir
kaynaklar / hedefler için anonimliğin korunması için onion route [11]. Nedeniyle
çok yollu yapısı, DBR algoritması doğal olarak ilgili gizliliği korur
herhangi bir ek gizlilik koruması kullanmadan aktarılan değer miktarı
teknikler (örneğin, ZKSNARK).
cOS. Zincir üzerindeki bir dApp, blok zincirine bağlanan bir ön uçtur. Zincir dışı
dApp'leri, yüksek ölçeklenebilirlik için büyük potansiyellere sahip olsalar da, oluşturmak o kadar
kolay değildir ve
geleneksel zincir üstü dApp'ler olarak kullanın. Celer Network, bir geliştirme olan cOS'u sunar.
Herkesin kolayca geliştirmesi, çalıştırması ve etkileşime girmesi için opment çerçevesi ve çalışma
zamanı
ek kompleks tarafından tıkanmadan ölçeklenebilir zincir dışı dApp'ler ile
zincir dışı ölçeklendirmeyle getirilenler. Celer Network, geliştiricilerin
uygulama mantığında ve en iyi kullanıcı deneyimini yaratın;
aşağıdaki görevleri içeren ağır kaldırma.
• Gelişigüzel zincir dışı ve zincir dışı durumlar arasındaki bağımlılığı bulun.
• Zincir dışı durumların takibi, depolanması ve anlaşmazlığının üstesinden gelin.
• Ara düğüm hatalarını şeffaf bir şekilde tolere edin.
• Birden çok eşzamanlı zincir dışı dApp'i destekleyin.
• Farklı zincir içi ve zincir dışı modüllere birleşik bir uygulama derleyin.
1.2. Celer Cryptoeconomics
Celer Network'ün kriptoekonomik mekanizması cEconomy,
damental ilke: iyi bir kriptoekonomik (belirteç) model, ek

Sayfa 10

değerler ve başka türlü imkansız olan yeni oyun-teorik dinamikleri mümkün kılar. Süre
ölçeklenebilirlik kazanan zincir dışı bir platform, ağ likiditesi konusunda da ödünleşmeler yapıyor
ve kullanılabilirliği belirtin ve bir kriptoekonomik model olmadan asla kalkmayacaktır.
yeni dinamiklerin bu ödünleşmeleri dengelemesini sağlayabilir.
Yeni değiş tokuşlar. Zincir dışı platform, aşağıdaki ödünleşmeleri yaparak ölçeklenebilirlik kazanır.
• Ölçeklenebilirlik ve Likidite. Zincir dışı değer transferi, mevduatın
ağ likiditesi olarak zincirde kilitlendi. Bu, özellikle potansiyel için zordur
zincir dışı hizmet sağlayıcıları, çünkü önemli miktarda likidite gereklidir.
küresel blok zinciri kullanıcıları için, giden olarak etkili zincir dışı hizmetler sağlamak
devlet kanallarında depozitolar veya yan zincirlerde dolandırıcılık cezası bonosu. Ancak sahipler
çok sayıda kripto varlığının (balinaların) ticari menfaati olmayabilir veya
zincir dışı bir hizmet altyapısı çalıştırmak için teknik yeterlilik,
Güvenilir ve ölçeklenebilir zincir dışı bir hizmet yürütme teknik yeteneğine sahip olmak

genellikle kanal mevduatları veya sahtekarlığa dayanıklı tahviller için yeterli sermayeye sahip değildir. Böyle
bir uyumsuzluk, kitlesel olarak benimsenmesi ve teknik evrim için büyük bir engel oluşturur zincir dışı işletim ağlarının.
• Ölçeklenebilirlik ve Kullanılabilirlik. Zincir dışı ölçeklendirme herhangi bir com-
blok zincirinin güvensiz özelliğine dair söz verirseniz, faydayı feda eder-
yetenek garantisi. Her eyalet kanalı veya zincir dışı sözleşme,
bir anlaşmazlık zaman aşımına uğrar ve ilgili taraf çevrimdışıyken risk altında olur
zaman aşımından daha uzun veya yerel durumlar kaybolduğunda.
Bu nedenle, yeterli miktarda likit sağlamak için teşvik uyumlu bir mekanizmaya ihtiyacımız var.
Güvenilir ve ölçeklenebilir bir zincir dışı hizmet yürütebilen kuruluşlar için
altyapı ve zincir dışı durumların her zaman pos-
sible on-chain anlaşmazlık.
Yeni kripto ekonomi. Zincir dışı ölçeklendirme çözümünü tamamlamak için,
vazgeçilmez getiren cEconomy adlı bir kriptoekonomik mekanizmalar paketi
değer verir ve ağ etkisi, istikrarlı likidite ve yüksek kullanılabilirlik sağlar.
Celer Network'ün protokol belirteci ("CELR") ve sıkıca bağlanmış üç bileşen.
• Likidite Taahhüdü Kanıtı (PoLC). PoLC sanal bir madencilik sürecidir
zincir dışı ekosistem için bol ve istikrarlı likidite elde eden. Eşit
ticipate, kişinin boştaki likiditesini (şu şekilde) taahhüt etmesi (kilitlemesi) yeterlidir.

Sayfa 11

kripto para birimleri ve CELR dahil ancak bunlarla sınırlı olmamak üzere dijital varlıklar
Teşvik olarak ödüllendirilen CELR ile belirli bir süre için zincir dışı platform
bu tür kullanıcılara.
• Likidite Destekleme Müzayedesi (LiBA). LiBA, zincir dışı hizmet sağlayıcılara olanak tanır
müzakere edilen faiz oranları ile “kalabalık kredilendirme” yoluyla likidite talep etmek.
Borç verenler belirlenen “mutluluk puanlarına” göre sıralanır.
faiz oranı, sağlanan likidite miktarı ve
hisseli CELR. Özellikle, daha fazla CELR hissesi olan kredi verenler (bunun bir göstergesi olarak)
ekosistemlere geçmiş katkıları) seçilmek için daha yüksek önceliğe sahiptir.
zincir dışı hizmet sağlayıcılara likidite sağlamak.
• Eyalet Koruyucu Ağı (SGN). SGN, özel bir kompakt yan zincirdir.
kullanıcıların çevrimdışı olduğu durumları korur, böylece kullanıcıların durumları her zaman
anlaşmazlık için uygun. Velilerin kazanmak için CELR'lerini SGN'ye yatırmaları gerekir
Kullanıcılardan fırsatları ve hizmet ücretlerini korumak.
Bölüm 5, CELR değerinin analizi ile birlikte cEconomy mekanizmalarını ayrıntılı olarak
tanıtmaktadır.
ve model teşvik uyumluluğu.
2. cChannel: Zincir Dışı Ölçeklemenin Temeli
Celer Network'ün cChannel'i, bir durum kanalı ve
maksimum esneklik ve verimlilikle yan zincir ağı. Bu bölüm şununla başlar:
genelleştirilmiş kanal, keyfi desteklemek için temel unsurları oluşturur ve ana hatlarıyla belirtir.
zincir üzerinde doğrulanabilir durumlar arasında koşullu bağımlılık. Sonra ufku genişletiyoruz
klasik devlet kanalının ötesinde ve yan zincirin nasıl aynı şekilde kapsülleneceğini inceleyin
üst katmana maruz kalan arayüz.
2.1. Genelleştirilmiş Devlet Kanalı
2.1.1. Anahtar Fikir ve Basit Bir Örnek
Mevcut ödeme ağı çözümlerinin önemli bir sınırlaması, destek eksikliğidir
genelleştirilmiş durum geçişleri için. Genelleştirilmiş durum geçişlerine duyulan ihtiyaç,
Ethereum gibi akıllı sözleşme platformlarının yükselişi. Akıllı sözleşme, eşzamansız
keyfi sözleşme mantığına dayalı kronik değer transferi. Ölçeklenebilirliği geliştirmek için

---

Sayfa 12

zincir dışı durum kanalı konseptlerini, zincir üstü durum geçişlerini kullanan bu tür blok zincirlerinin
zincir dışı durum kanallarına konulmalı ve karşılık gelen değer aktarımı
bu tür durum geçişlerinden haberdar olunmalıdır.
Temel fikri açıklamak için basit bir koşullu ödeme örneği kullanıyoruz.
zincir üzerindeki durumların zincir dışı durum geçişlerine nasıl dönüştürüleceği. Diyelimki
Alice ve Carl, böyle bir oyunun sonucuna bahis yaparken bir masa oyunu oynamak istiyorlar.
güvensiz bir şekilde: Alice kazanırsa Carl'a 1 $ ödeyecek ve tam tersi.
Bu, zincir üzerinde uygulamak için basit bir mantıktır. Akıllı bir sözleşme yapılabilir
oyun başlamadan önce Alice ve Carl'ın ödemesini tutar. Alice ve Carl olacak
Zincir üzerindeki akıllı sözleşmenin işlevlerini çağırarak sadece bu oyunu oynayın. Biri
oyunu kaybeder, teslim olur veya zaman aşımına uğrar, kazanan kaybeden depozitoyu alır.
Mevduatlar, şartlı olarak düzenlenen ödemeler olarak görülebilir (örn.
karşı taraf kazanır). Ne yazık ki, zincir içi akıllı sözleşme işlemleri
her işlem bir zincir içi işlem içerdiğinden son derece yavaş ve pahalıdır.
Zincir dışı durum kanalı, bakım sırasında ölçeklenebilirliği önemli ölçüde iyileştirmek için
kullanılabilir.
aynı semantik lekelemek. Alice ile arasında bir ödeme kanalı olduğunu varsayalım.
Carl. Yukarıdaki semantiği etkinleştirmek için, kanalın işlevselliğini genişletmemiz gerekir.
Durum kanıtı, oyunun kazanan durumuna bağlı olan koşullu bir kilit içerecek şekilde.
Alice daha sonra Carl'a zincir dışı bir koşullu ödeme göndererek etkin bir şekilde şunu söyleyerek: "
Oyun sözleşmesi kazanan işlevi Carl'ın kazandığını söylüyorsa, Carl'a 1 $ ödeyin ". Oyun
durum geçişleri de zincir dışına taşınabilir. En basit yol, hala
masa oyununun kuralını düzenleyen bir zincir içi sözleşmeye sahip olmak ve bu sözleşmenin
adres, şartlı ödemede belirtilmiştir. Tüm durum geçişleri olur-
Zincir üzeri içerisine enjekte edilebilen karşılıklı olarak imzalanmış oyun durumları aracılığıyla zincir dışı oluşturma
gerektiğinde sözleşme yapın.
Ama aslında, program için herhangi bir değer bağına gerek olmadığı için
devletler, tüm oyun sözleşmesi ve ilişkili durumlar her zaman zincir dışı kalabilir.
ilgili taraflar işbirliği içinde olduğu sürece. Tek şart, ilgili
oyun durumları gerektiğinde zincir üzerinde doğrulanabilir. Zincir üzerinde doğrulanabilir bir durum,
diğer sözleşmeler veya nesneler belirsizlik olmadan ona atıfta bulunabilir. Bunu anlamak için ihtiyacımız var
zincir dışı referansları (karma gibi) eşleyen bir referans çevirmen sözleşmesine sahip olmak
sözleşme kodu, yapıcı parametreleri ve nonce) zincir içi referanslara (sözleşme
adres). Bu yapılarla, Alice ve Carl arasındaki oyun yalnızca bir

---

Sayfa 13

Oyun mantığına özgü olmayan ve zincir üzerinde olmayan uzun vadeli zincir içi sözleşme
oyun için çalıştırma veya başlatma.
Yukarıdaki örnek, özel ve basit bir zincir dışı tasarım örneğini yansıtmaktadır.
desenler ve çok daha karmaşık olabilir. Koşullu ödeme olabilir
basit Boole koşullarından daha karmaşıktır ve bir
keyfi sözleşme mantığına dayalı olarak kilitli likiditeyi yeniden dağıtmanın yolu. Aslında,
koşullu ödemeler, daha genelleştirilmiş bir koşullu durumun basit bir özel durumudur
geçiş. Kanal bağımlılığı da bire bir olmaktan daha karmaşık olabilir
çok sekmeli durum rölelerinin ortak modelini gerçekleştirmek için bağımlılık. Detaylandırıyoruz
aşağıdaki bölümlerdeki teknik özellikler.

2.1.2. Tasarım Hedefleri

En büyük hedefimiz hızlı, esnek ve güven içermeyen zincir dışı etkileşimler sağlamaktır. Bekliyoruz
çoğu durumda zincir dışı durum geçişleri, nihai çözüme kadar zincir dışı kalacaktır. Orada-

dolayısıyla, yaygın olarak kullanılan zincir dışı kalıpları kısa ve öz etkileşimlere dönüştürmeyi hedefliyoruz.
zincir üzerindeki bileşenlerden yerleşik destek ile.
İkinci hedefimiz, çalışan veri yapısı ve nesne etkileşim mantığını tasarlamaktır.
farklı blok zincirleri için. Celer Network, blok zincirinden bağımsız bir platform oluşturmayı hedefliyor
ve akıllı sözleşmeleri destekleyen farklı blok zincirleri üzerinde çalışmak. Bu nedenle, ortak bir veri yapısı şeması ve belirli bir yönlendirme katmanı gereklidir.
Vurgulanan bu iki hedefin yanı sıra, değişikliklerin resmi özelliklerini kullanmayı planlıyoruz.
nel durum makineleri ve iletişim ile birlikte güvenlik özelliklerini doğrulayın
bu durumları değiştiren protokoller. Ayrıca verimli bir zincir içi sağlamayı da hedeflemeliyiz
mümkün olduğunda çözüm mekanizması.
2.1.3. Genel Özellikler
Bu bölümde, cChannel'in gen-
Yukarıdan aşağıya bir yaklaşımla eralize devlet kanalı ve Ortak Durumu açıklar
Değer aktarımı ve arbi ile herhangi bir durum kanalı için geçerli olan Kanal Arayüzü
üçlü sözleşme mantığı. Aşağıdakiler için kapsamlı uzmanlık ve optimizasyon olabilir
farklı somut kullanım durumları, ancak ilkeler aynı kalır.
Genelleştirilmiş bir devlet kanalının ayrıntılı spesifikasyonundan önce, ilk olarak
Bu bölümde kullanılacak birkaç önemli notasyon ve terim.

Sayfa 14

• (Durum). Bir kanalın durumu ile ifade edin. İki taraflı bir ödeme kanalı için,
iki tarafın mevcut bakiyelerini temsil eder; bir masa oyunu için, s temsil eder
yönetim kurulu durumu.
• (Eyalet Kanıtı). Durum kanıtı, zincir üzeri veri yapısı arasında bir köprü görevi görür.
sözleşmeler ve zincir dışı iletişim protokolleri. Durum kanıtı sp,
aşağıdaki alanlar
sp = {Δs, seq, merkle kökü, sigs},
(1)
burada, şimdiye kadarki birikimli durum güncellemelerini gösterir. Verildiğine dikkat edin
temel durum 0 ve bir durum güncellemesi, benzersiz bir şekilde yeni bir kanal durumu oluşturabiliriz
s. Örneğin, iki taraflı bir ödeme kanalında temel durum s 0 ,
iki tarafın mevduatları ve durum güncellemeleri,
bir katılımcıdan diğer katılımcıya aktarılan jeton miktarı. seq şudur:
durum kanıtı için sıra numarası. Daha yüksek sıra numarasına sahip bir durum ispatı
daha düşük sıra numaralı durum provalarını devre dışı bırakır. merkle kökü,
tüm bekleyen koşul gruplarının merkle ağacı ve koşullu
cChannel'daki durumlar arasında bağımlılık. Son olarak, işaretler,
Bu devlet kanıtı üzerindeki tüm taraflar. Devlet kanıtı yalnızca tüm tarafların imzaları geçerliyse geçerlidir.
mevcut.
• (Durum). Koşul koşul, temel birimi temsil eden veri yapısıdır.
koşullu bağımlılık ve bu, koşullu bağımlılık DAG'lerinin
dokuma. Aşağıdaki gibi bir koşul belirtilebilir.
koşul = {zaman aşımı, ∗ IsFinalized (bağımsız değişkenler), ∗ QueryResult (değiştirgeler)}
(2)
Burada zaman aşımı, koşulun sona erdiği zaman aşımıdır. Örneğin, bir
bir tahta oyununun sonuçlarına bağlı olan durum, zaman aşımı şuna karşılık gelebilir:
masa oyununun maksimum süresi (örneğin, onlarca dakika). Boole işlevi
işaretçi IsFinalized (değiştirgeler) koşulun çözülüp çözülmediğini kontrol etmek için kullanılır
ve koşul zaman aşımından önce ayarlandı. Bu işlev çağrısı için argümanlar
uygulamaya özel. Örneğin, masa oyununda argümanlar basit olabilir.

ple as args = [blocknumber] oyunun kazananının belirlenip belirlenmediğini sorgulama
13

Sayfa 15

blok numarasından önce. Ek olarak, QueryResult (args) bir sonuç sorgu işlevidir
koşulun çözümleme sonucu olarak gelişigüzel baytlar döndüren işaretçi. Sınav için-
ple, masa oyununda, argümanlar args = [player1] olabilir mi?
player1 kazanır (boole koşulu); ikinci fiyat açık artırmasında, tartışma
ments = [katılımcı1, katılımcı2, ⋯, katılımcıN] kimi sorgulayan olabilir
kazanan ve her katılımcının ödemesi gereken para miktarıdır (genel koşul
yon). Bir koşul için çözüm süreci ilk olarak IsFinalized (args) gerçekleştirmektir.
ve sonra QueryResult (bağımsız değişkenler) sonuç sorgusu gerçekleştirin.
• (Koşul Grubu). Koşul grubu koşul grubu, aşağıdakiler için daha üst düzey bir soyutlamadır:
genelleştirilmiş durum bağımlılıklarını ifade etmek için bir grup koşul. Bir koşul grubu
aşağıdaki gibi belirtilebilir.
koşul grubu = {Λ, ResolveGroup (koşul sonuçları)},
(3)
burada Λ, bu koşul grubunda yer alan bir dizi koşulu belirtir. Her biri
koşul koşul ∈ Λ keyfi bir bayt dizisine çözümlenir (yani,
cond.QueryResult (değiştirgeler)). Bu bayt dizisi, bir grup çözümleme tarafından ele alınır.
tüm koşulların çözümleme sonuçlarını alan ResolveGroup (koşul sonuçları) işlevi
girdi olarak seçer ve bir durum güncellemesi döndürür. Bir ödeme kanalı için, her bir
ödeme grubu, şartlı bir ödemeye karşılık gelir. Örneğin, şartlı ödeme
"B, Gomoku oyununu kazanırsa A, B $ 1 öder" ifadesi,
iki koşulu içeren koşul grubu: Karma Zaman Kilidi koşulu
(çoklu atlama geçişi için) ve Gomoku oyun koşulu ("B oyunu kazanır").
ResolveGroup işlevi, her ikisi de
koşullar doğrudur.
Artık bir durum kanalı için arayüz belirlemeye hazırız. Durum kanalı C olabilir
aşağıdaki demet olarak belirtilmelidir:
C = {p, s 0 , sp, s, F, τ},
(4)
p = {p 1 , p 2 , ..., p n } bu kanaldaki katılımcı kümesidir. s 0 , zincir üstü tabanıdır
bu kanal için durum (örneğin, bir ödeme kanalındaki her katılımcı için ilk depozitolar).
sp, kanal için bilinen en güncel durum kanıtını temsil eder. güncellendi
14

Sayfa 16

durum kanıtı sp tamamen yerleştikten sonra kanal durumu. τ, yerleşim zaman aşımı artışıdır
daha sonra belirtilecek olan devlet kanıtı için. F bir dizi standart işlev içerir
her eyalet kanalı tarafından uygulanmalıdır:
• ResolveStateProof (sp, koşul grupları). Bu işlev mevcut durumu günceller
ekli koşul gruplarını çözerek kanıt.
• GetUpdatedState (sp, s 0). Bu işlev, en güncel durumu elde etmek için kullanılır
zincir dışı durum kanıtı sp ve zincir üstü temel durumlara dayanır s 0 .
• UpdateState (ler). Bu işlev, halihazırdaki durum kanalının zincir üzerinde güncellemelerine izin
verir.
çözülmüş durum s.
• IntendSettle (yeni sp). Bu işlev, hesaplaşmadan önce bir meydan okuma dönemi açar.
ment zaman aşımı. Zorluk süresi boyunca, bu işlev girdi olarak bir durum kanıtı alır
ve giriş daha yeniyse mevcut durum kanıtını güncelleyin.
• ConfirmSettle (sp). Bu işlev mevcut durum kanıtını doğrular ve onaylar
mevcut zamanın yerleşim zaman aşımını aştığı göz önüne alındığında tamamen yerleşmiş olarak.
• IsFinalized (args) ve QueryResult (args), con-

ikili bağımlılık. Dış sorguları, gerekli argümanlar ile kabul eder.
buna göre yorumlamak için sözleşme sorgulama. Aslında, bazı kalıplar sıklıkla kullanılır
yeterli, cChannel'in uygulamasında, onları önceden tanımlanmış işleve ayırıyoruz
arayüzler.
• CloseStateChannel (ları). Bu işlev, durum değişiminin yaşam döngüsünü sonlandırır.
nel ve en son yerleşik devletlere göre gerekli durumları dağıtır.
Yerleşim zaman aşımı, son çağrılan ResolveStateProof zamanına göre belirlenir
veya SettleStateProof ve yerleşim zaman aşımı artışı τ.
Bağımlılık Kısıtlamaları. Farklı durumlar arasında bağımlılıklar yarattığımızda
uygun bir çözümü garanti etmek için bazı kısıtlamaların uygulanması gerekir.
DAG bağımlılığının olusyonu. Durum kanalı C 1'in durum kanalına bağlı olduğunu varsayalım
C 2 . Daha sonra C 1 katılımcılarının katılımcının bir alt kümesi olması gerekir.
C 2'nin pantolonu öyle ki C 1 katılımcılarının çözme konusunda gerekli bilgilere sahip olması
C 1'e bağımlılığı .

Sayfa 17

2.1.4. Ortak Araçlar
Yukarıdaki soyutlama, genelleştirilmiş durum kanalı konvansiyonu için ortak modeli tanımlar.
yapı. Farklı blok zincirlerinde, gerçek uygulama farklı olabilir. İçin
Örneğin, Ethereum'da, çapraz sözleşme aramaları dönüş değeri içerir, ancak Dfinity'de çapraz
sözleşme aramaları yalnızca kayıtlı geri aramaları tetikler. Birden fazla blok zincirinin uygulanmasının
gözden geçirilmesi
durum geçiş sanal makinelerinden bahsederken, gerekli olan iki ortak yardımcı programı belirledik
genelleştirilmiş devlet kanalının pratikte işleyişi için aşağıdaki gibi:
• Zincir dışı Adres Çeviricisi (OAT). Yukarıdaki soyutlamada, durum ve
durum grubu farklı işlevlerle ilişkilendirilir. Bu işlevler olmalıdır
Zincir üzerindeki sözleşmenin işlevlerinin referansı, ancak programdan beri (akıllı sözleşme)
durumlar doğal olarak blok zincirindeki kısıtlamalara bağlı değildir,
zincir üzerinde bir varlığa sahip olmak için temel gereksinim. Hareket etmenin önündeki tek engel
bunlar tamamen zincir dışıdır, aşağıdaki gibi işlevler için olası referans belirsizliğidir.
IsFinalized ve QueryResult.
Bu belirsizliği çözmek için, zincir dışı eşlemeyi sağlamak için bir zincir içi kural kümesi
tanımlayabiliriz.
zincir üzerindeki referanslara referanslar. Off-Chain Address Translator bunun için tasarlanmıştır.
Değer içermeyen bir sözleşme için, benzersiz bir tanımlayıcıyla referans gösterilebilir
sözleşme kodu, başlangıç durumları ve belirli bir nonce aracılığıyla oluşturulur. Biz ararız
böyle benzersiz tanımlayıcı zincir dışı adres. Zincir üzerindeki koşulları çözerken,
başvurulan sözleşmelerin ve ilgili işlevlerin (örneğin,
IsFinalized ve QueryResult) zincir dışı adresleri
zincir üstü adresler. Bu tür bir işlevselliği gerçekleştirmek için, OAT'nin
Zincir içi sözleşmeyi almak ve kurmak için sözleşme kodu ve zincirdeki ilk durumlar
zincir dışı adresten zincir üzerindeki adrese eşleme.
• Hash Time Lock Kaydı (HTLR). Hash Time Lock yaygın olarak
birden çok devlet kanalını içeren işlemlerin gerçekleşmesi gereken senaryo
atomik olarak. Örneğin, çok sekmeli aktarmalı ödeme (koşulsuz veya koşullu),
farklı belirteçler, çapraz zincir köprüleri ve daha fazlası arasında atomik takas. HTL olabilir
Tamamen zincir dışı uygulandı, ancak Sprite [5] 'in belirttiği gibi, bu bir aşırı
aslında zincir dışı ölçeklenebilirliği sınırlayan optimizasyon. Bu nedenle, Sprite [5] pro-
tüm kilitlerin başvurabileceği merkezi bir kayıt defteri oluşturur. Sprite'ı genişletip değiştiriyoruz:
cChannel'in ortak modeline uyar. Etkili bir şekilde, HTLR bağımlılık uç noktaları sağlar
(IsFinalized, QueryResult) kilit görevi gören koşullar için. IsFinalized çekimler

---

Sayfa 18

bir karma ve blok numarası ve karşılık gelen ön görüntü varsa true döndürür.
blok numarasından önce kaydedilir. QueryResult bir hash alır ve eğer
karmanın ön görüntüsü kaydedilir. Bu iki işlev daha da basitleştirilebilir
bire, ancak genellik adına, onları iki ayrı olarak tutabiliriz.
fonksiyonlar. HTLR'nin ve ilişkili IsFinalized ve QueryResult'un
her zaman zincirde.

2.1.5. Kutudan Çıkmış Özellikler

Ek olarak, yaygın olarak kullanılacak ve geliştirilecek kalıplara bakmamız gerekiyor.
karşılık gelen kapalı devreyi basitleştirmek için kullanıma hazır özelliklere sahip belirli zincir üstü bileşenler
zincir etkileşimleri. Genelleştirilmiş Ödeme Kanalı (GPC) çok iyi bir örnektir
bunun. Genelleştirilmiş Ödeme Kanalı, genel kullanıma uygun ödeme kanalıdır.
kanal özelliklerini belirtir ve bu nedenle çeşitli şartlı ödemeleri destekleyebilir
diğer zincir içi veya zincir dışı nesnelere dayanır.
Öncelikle soyut modeli GPC bağlamında daha somut hale getiriyoruz. s 0 temsil eder
her bir taraf için statik mevduat haritası s. s her biri nihai netleştirilmiş değeri temsil eder
parti sahibi. SubmitStateProof, bir durum kanıtı ve tetikleyici gönderme işlevidir.
SettleStateProof çağrılmadan önce bir zaman aşımı sorgulama süresi ve
devlet kanıtı. IsFinalized ve QueryResult, durumun
Bu ödeme kanalının% 'si kesinleşti ve cari bakiyeleri sorguladı. Bir mayıs
bir ödeme kanalının neden dış sorgu için bir arayüze sahip olduğunu merak ediyorum. Bunun nedeni bazılarının
diğer ödemeler veya durumlar belirli şartlı ödemelerin varlığına veya varlığına bağlı olabilir
sp kilitli. ResolveStateProof en ilginç kısımdır çünkü
özel optimizasyon gerçekleşecek ve zincir dışı etkileşimi büyük ölçüde azaltacaktır
karmaşıklık. GetUpdatedState, netleştirilmiş olanı hesaplamak için basit bir işlevdir.
ilk depozito ve tamamen çözülmüş sp temel alınarak her bir taraf için ödeme.
CloseStateChannel, basitçe kanalı kapatır ve netlenmiş olanı dağıtır.
her parti için nihai denge.
Bu temel modelle, GPC yapılarını nasıl daha da optimize edebileceğimizi tartışıyoruz
kullanıma hazır özellikleri etkinleştirmek için.

• İşbirliğine Dayalı Yerleşim Çoğu durumda, devlet kanalı uygulamalarının karşı tarafları
kooperatiftir. Sonuç olarak, daha karmaşıklık ve maliyetten geçme
meydan okuma süresi ve sonra yerleşir. Bu nedenle, cChannel ortaklaşa yerleşime olanak sağlar

---

Sayfa 19

karşı tarafların yalnızca en son devlet kanıtını değil, aynı zamanda
sonuçta açıklanan durum güncellemesinin anlaşmayı göstermek için ortaya çıkan imza
devlet kanıtı gerçekten de son durumdur. Bununla, kapatılacak işlem sayısı
bir durum kanıtı 2'den 1'e düşebilir.

• Tek İşlemli Kanal Açma cChannel'in getirdiği diğer bir optimizasyon
bir kanalı açmak için zincir üzerinde işlem sayısını 3'ten 1'e düşürmek için. Bu,
karşı taraflar için mevduatları depolamak için bir bağımlılık sözleşmesi kullanılarak elde edilir. The
$N-1$ karşı taraf, tamamen zincir dışı para çekme yetkisini imzalayacak
ve bir karşı taraf, kanal açılışını tamamlamak için bunu zincir üzerinde gönderebilir
süreç.

• Doğrudan Nihai Durum Talebi Genelleştirilmiş durum kanalı uygulamaları oluştururken,
koşullu durum bağımlılığı yaygın olarak kullanılmaktadır. GPC'yi sonlandırırken, bir taraf
koşullu bağımlılık grafiğini geçme sürecinden kaçınmak isteyebilir. Bu
karşı tarafın gittiği karşı tarafın acı senaryosunu sınırlamaktır.
çevrimdışı ve bazı Koşul Gruplarını işbirliği içinde koşulsuz olarak dönüştürmeyi reddetme
durum güncellemesi. İhtilaflı taraf için gereken çalışmayı sınırlandırmak için,

doğrudan nihai durum iddiası yöntemi. Çevrimiçi tarafın doğrudan nihai bir nihai talep etmesine izin verir.
aslında herhangi bir ek bağımlılık grafiği geçişi gerçekleştirmeden durumu.
Karşı taraf imzasına gerek yoktur. Kötüye kullanımı önlemek için sahtekarlığa dayanıklı bir bağ da hak talebinde bulunan taraf için gereklidir. Zorlu bir dönemden sonra devlet,
herhangi bir ek işlem gerçekleştirmeye gerek kalmadan nihai.
• Dinamik Para Yatırma ve Çekme. GPC için ortak bir gereksinim şudur:
karşı taraf karşılıklı olmadığında sorunsuz zincir içi işlemleri etkinleştirmek için
ağa bağlı. Para çekme işlemleri için bir çift işlev sunuyoruz.
Bu gereksinimi karşılamak için IntendWithdraw ve ConfirmWithdraw.
IntendToWithdraw , bir meydan okuma süresi ile temel durumu s 0 değiştirir . Sayaç
taraf, ihtilaf için çelişkili sp sunabilir. Zorluktan önce bir anlaşmazlık olmazsa-
τ tarafından tanımlanan lenge dönemi, onaylamak ve iptal etmek için ConfirmToWithdraw çağrılır
çekilme haraç. Bu iki işlev, IntendSettle'a çok benzer şekilde çalışır.
ve ConfirmSettle. Para yatırma işlemi basittir çünkü yalnızca temel durumu değiştirir
s 0 .
• Boolean Devre Koşul Grubu. GPC'nin en yaygın kullanımının
durum, Boole devresine dayalı koşullu ödeme olacaktır. Örneğin, "A, B'yi öder, eğer

---

Sayfa 20

X işlevi veya Y işlevi true döndürür ”. Bu tür bir ödemeyi optimize etmek için,
koşul grubu ve koşulun arayüzü. Özellikle, işlevi uzmanlaştırabiliriz
ResolveGroup, koşullardan herhangi biri varsa önceden tanımlanmış bir koşullu ödemeyi serbest bırakacak
sonuçların (veya koşul sonuçlarının herhangi bir boole devresinin) çözümlenmesi doğrudur. Bu yoldan,
ResolveGroup için ek nesneler oluşturma zahmetinden kurtulduk ve
karşılık gelen çok taraflı iletişim ek yükü. Koşulu şu şekilde de belirtiyoruz:
Boolean koşulu, böylece bağlı nesnelerin bir arayüze sahip olmasını gerekli kılıyoruz
sorgulanan duruma göre doğru veya yanlış döndüren "isSatisfied" etkisiyle.
• Fon Tahsis Koşul Grubu. Daha genelleştirilmiş başka bir kullanım durumu
GPC, genelleştirilmiş durum atamasıdır. Bunu başka bir
içinde yalnızca bir koşul bulunan farklı türde koşul grubu. QueryResult
doğrudan Δs için bir güncelleme dikte eden bir durum atama haritası döndürür. Bu olanak sağlar
GPC için daha genel bir eklenti noktası. Zincir dışı bir sözleşme eklenebilir.
likidite belirli bir şekilde kilitlendi. Bu sözleşme sadece kontrol edemez
bir oyunu kim kazanır (Boolean), ancak kazananın oyunu kazanmak için kaç adım attığı,
ve sonra belirli bir hesaplama yaparak likiditeyi tayin edin. İlgili
taraflar, kontrol fonksiyonuna referans veren bir Koşul Grubu oluşturabilir.
karşılıklı olarak üzerinde anlaştıkları zincir dışı sözleşme adresi.
Farklı modeller için tanımlanmış daha birçok ortak model olabilir, ancak
Yukarıdaki örnek, bu tür bir optimizasyon için tasarım ilkesini göstermektedir.

2.2. Yan Zincirli Alternatif Kanal Modeli

Yukarıda bahsedilen genelleştirilmiş durum kanal modelinin yanı sıra, cChannel ayrıca
yan zincirler [1] tarafından kolaylaştırılan alternatif bir durum kanalı modelini oluşturur. İçin
Örneğin, birden çok kullanıcının birbirine ödeme yapması gereken senaryoyu düşünün. Kullanıcılar şunları yapabilir:
mevduatlarını, bir yan zincir sözleşmesi gibi hareket eden merkezi bir sözleşmeye havuzlamak
Blok teklif verenlerin rolünü oynayan zincir dışı hizmet sağlayıcıları (bir "çok taraflı
zincir dışı hizmet sağlayıcılarının "merkez operatörleri" olduğu hub ") ve bu nedenle
bir merkezde birden çoğa ödeme ilişkileri. Zincir dışı hizmetin bütünlüğü
Sağlayıcılar, katılımcılar tarafından kabul edilebilir belirli bir dolandırıcılık geçirmez bağ ile sağlanır.
Spesifik olarak, Celer Network'te her zincir dışı servis sağlayıcı bir yan zincir çalıştırabilir.

Sayfa 21

destekli devlet kanalı:
C = {s, p ∗
, b, τ},
(5)
s yan zincir durumu olduğunda, p ∗ tek blok önericidir (zincir dışı hizmet
sağlayıcı), b, dolandırıcılık bağıdır ve τ, kesinlik zaman aşımını temsil eder. Yapabildiğim her düğüm
En son güncellemeyi güncellemek için kanaldaki diğer tüm düğümlere yan zincir işlemleri gönderin
durum. Herhangi bir yan zincir işleminde olduğu gibi, node i sadece bu işlemi imzalamayacağım
aynı zamanda bu işlemin dahil edildiğini kanıtlamak için başka bir işlem imzalayın.
p ∗ tarafından oluşturulan blok . İkinci imzalı işlem, düğümün kanıtı olarak kullanılabilir
ben. Katılımcıların blok verileri tamamen mevcut olduğu sürece, bu yan zincirin son hali
işlem de hızlı bir şekilde onaylanabilir.
Bu yan zincir destekli kanal modeli, aşağıdaki beklenen faydalarla birlikte gelir [1]
daha önce bahsedilen kanal modelleriyle karşılaştırıldığında.
• Alıcı için zincir üzerinde işlem ve çevrimiçi varlık gerekmez.
Bu, yan zincir özelliklerinden miras alınan doğal bir avantajdır. Sebep şu ki
alıcı, yan zincir destekli kanaldan aldığı parayı -
aslında herhangi bir yan zincir depozitosunu kendisi gerçekleştiriyor.
• Parti başına fon kilitlemesi yok. Bu fayda, ödeme değişikliği bağlamında
nels. Çok taraflı ödemeler için yan zincir tabanlı kanallar kullanıldığında,
Tarafın birbirlerine ödeme yapmadan önce depozitolarını önceden kilitlemesi gerekmez
(sahtekarlığa dayanıklı tahvil yatırması gereken blok teklifçisi hariç).
Bununla birlikte, ekosistem bu kanalın dezavantajlarının açıkça farkında olmalıdır.
aşağıdaki gibi model:
• Dolandırıcılığa dayanıklı tahvil hala gereklidir. Yan zincir tabanlı kanallar söz konusu olduğunda,
Dolandırıcılığa dayanıklı tahvil, blok teklif edenlerden doğrudan p ∗ tarafından hala gereklidir
veya denetim ve sigorta hizmetleri sağlayan kişi. Açıkça olmalı
blok önericisi için en kötü durum likidite gerekliliğinin (yani,
zincir dışı hizmet sağlayıcı) aslında sınırsızdır. Nedeni şu ki
Yeterince işbirliği, kötü niyetli taraf sınırsız tekrarlanan harcama yaratabilir.
• Veri kullanılabilirliği sorunu kesinliği karmaşıklaştırabilir. Kötü niyetli taraf olmasa bile
işin içine girerse, yan zincir modelinin doğasında var olan kesinlik gecikmesi hala bu

Sayfa 22

kanal modeli, özellikle veri kullanılabilirliği bir sorun haline geldiğinde. Ne zaman
blok verileri her zaman ilgili taraflar arasında mevcut değildir, yan zincir yüzleri
kaçınılmaz yeniden organizasyon ve bu nedenle kesinlik en iyi şekilde ertelenecek ve
en kötü durumda tüm yan zincir terk edilecektir.
Bu yan zincir tabanlı kanallar, şu yolla birbirine bağlanabilir:
ortak devlet kanalları.

3. cRoute: Provably-Optimal Value Transfer Routing

3.1. Eyalet Kanalı Ağ Yönlendirmesindeki Zorluklar

Durum yönlendirme ihtiyacı (veya ödeme kanalı durumunda "ödeme yönlendirme")
ağlar) aşikardır: her biri arasında doğrudan devlet kanalları kurmak pratik değildir.
kanal açma maliyetleri ve mevduat likidite kilidi nedeniyle düğüm çifti. Bu nedenle
durum geçişlerinin olduğu durum kanallarından oluşan bir ağ oluşturmak için gereklidir.
güvensiz bir şekilde iletilmelidir. Durum yönlendirmesinin tasarımı,
bir eyalet kanalı ağının sağlayabileceği ölçeklenebilirlik düzeyi, yani ne kadar hızlı ve nasıl
belirli bir ağ üzerinde birçok işlem akabilir. Ancak, mevcut tekliflerin tümü düşüyor
devletin benzersiz özelliklerinin dayattığı temel zorlukları karşılamak için kısa
kanal ağları.

Merkezi olmayan ödeme için bir seçenek olarak Landmark yönlendirme [16] önerildi
çeşitli ödeme kanalı ağlarında yönlendirme. Örneğin, Lightning Network [9]
Flare [10] adlı bir dönüm noktası yönlendirme protokolünü benimser. Benzer bir algoritma da kullanılıyor
merkezi olmayan IOU kredi ağında SilentWhispers [4]. Dönüm noktasıyla ilgili temel fikir
yönlendirme, bir ara aracılığıyla gönderenden alıcıya en kısa yolu belirlemektir.
bir dönüm noktası olarak adlandırılan diat düğümü, genellikle yüksek bağlantıya sahip iyi bilinen bir düğüm.
Raiden Network [2] (bir ödeme kanalı ağı) birkaç uygulamadan bahsetti
A gibi ödeme yönlendirme için alternatifler * Bir im-dağıtılır ağaç arama
en kısa yol yönlendirmesinin plementasyonu. Ek olarak, rota keşfi zor olduğu için ancak
düğümler, bazı kolaylık sağlamak için diğer düğümler için yol bulma hizmetleri sağlayabilir
Raiden Ağındaki ücretler.
Yakın zamanda önerilen SpeedyMurmurs [13], önceki en kısa yol yönlendirmesini geliştirir
algoritmaları (Lightning Network ve Raiden Network'te kullanıldığı şekliyle) hesaba katarak
her bir ödeme kanalındaki mevcut bakiyeler. SpeedyMurmurs, özellikle

---

Sayfa 23

Bir
C
B
100
100
100
100
100
100
Bir
C
B
Bir
C
B
200
0
0
200
200
0
Bir
C
B
Bir
C
B
0
200
200
0
0
200
Bir
C
B

zaman dilimi 1
zaman dilimi 2
zaman dilimi 3
Şekil 2. En kısa yol yönlendirme, kanal dengesizliği nedeniyle sık topoloji değişikliklerine neden olur.
P2P ağlarında yaygın olarak kullanılan gömme tabanlı yönlendirme algoritmaları
[12], önce bir önek ağacı oluşturur ve ardından her düğüm için bir koordinat atar.
Her ödemenin iletilmesi, koordinat arasındaki mesafelere bağlıdır.
o düğüm ve hedefin koordinatı tarafından biliniyor. Önek ağacı ve
Kaldırılması gereken herhangi bir bağlantı varsa, her düğümün koordinatı ayarlanacaktır
(yani, bir bağlantının dengesi tükendiğinde) veya eklendiğinde (yani, tükenen bir bağlantı yeni bir bağlantı aldığında
finansman).
Mevcut tüm yönlendirme mekanizmalarının "kısa" duruma geldiği gözlemlenebilir.
Mevcut bakiye bedeli ile tahmini yol yönlendirme ”. Geleneksel veri ağlarında,
en kısa yol yönlendirme, makul ölçüde iyi bir iş hacmi ve gecikme performansı sağlar.
Mance, ağ topolojisinin nispeten sabit kaldığı varsayımına dayanarak ve
bağlantı kapasitesi "vatansız" dır (yani, her bağlantının kapasitesi geçmiş işlemlerden etkilenmez.
misyonlar). Ne yazık ki, böyle bir varsayım artık zincir dışı bir durum için geçerli değil
"durum bilgisi olan" bağlantı modeli nedeniyle kanal ağı, yani kapasite (kullanılabilir denge)
Her yönlendirilen bağlantının% 50'si, ödemeler bu bağlantıdan geçerken değişmeye devam eder. Bunu not et
en kısa yol yönlendirmesi, kanal dengelemesini hesaba katmaz ve bu nedenle her bağlantı,
Kapasitesinin hızla tükenmesi, ağda sık sık değişikliklere neden olur
topoloji. Şekil 2, en kısa yol yönlendirmesinin topol'a yol açtığı bir senaryoyu göstermektedir.
ogy her zaman aralığında değişir. Her zaman aralığının başında düğümün
A, düğüm B ve düğüm C'nin her biri, düğüm B, düğüm C'ye 100 jetonluk bir ödeme başlatır.
ve düğüm A, sırasıyla. İlk kanal bakiyesi dağılımı altında (zaman aralığı
1), her düğüm çifti çift yönlü bir bağlantıyla bağlanır ve her düğüm,
en kısa yol yönlendirmesi altında hedefine doğrudan bir yol. Ancak bu sonuç

Sayfa 24

her kanal üzerinden tek yönlü bir transferde ve dolayısıyla kanalın dağıtımında
temel topolojinin bir
saat yönünün tersine döngü. Bu yeni topolojide, en kısa yol yönlendirmesi yapmaya devam ediyor
tek yönlü transferler (örneğin, A → B ödemesi için A → C → B yolunu seçer) ve
kanal bakiyeleri, temel topolojinin olduğu diğer uca itilir.
tamamen saat yönünde çevrildi (zaman dilimi 3). Aynı kalıp tekrar edecek
süresiz. Bunun aksine, C düğümü daha uzun bir C → B → A rotasını takip ederse, her kanal
her zaman dengeli kalacak ve ağ topolojisi asla değişmeyecektir. Herhangi
en kısa yol yönlendirmesinin merkezi olmayan uygulaması, bu tür sık topoloji değişiklikleri
algoritmanın yakınsaması zaman aldığından düşük performansa yol açabilir
yeni topoloji (örneğin, SpeedyMurmurs [13] 'te olduğu gibi önek ağacını yeniden yapılandırmak için), dur-
hangi optimal altı yolların izlenebileceğini belirlemek. Daha da kötüsü, ağın
Topoloji, algoritma yakınsamadan önce tekrar değişebilir ve dolayısıyla algoritma
asla yakınlaşmayabilir ve sürekli olarak düşük iş hacmi performansına ulaşmayabilir.
En son Revive [3] projesinin açık bir kanal yeniden dengeleme önerdiğine dikkat edin
düzeni. Bununla birlikte, Revive durum yönlendirmesini hesaba katmaz, bu da onun
kanal yeniden dengeleme prosedürü, temeldeki yönlendirme sürecine şeffaf değildir ve
ekstra bant dışı koordinasyon gerektirir. Dahası, Revive yalnızca kısıtlı bir
döngüsel yapılar içeren ağ topolojileri sınıfı ve herhangi bir
Kanal yeniden dengeleme prosedürünün genel bir topolojide uygulanabilir olduğunu garanti eder. İçinde
karşılaştırma, şeffaf ve en uygun hale getiren bir yönlendirme algoritması öneriyoruz.

yönlendirme işlemi sırasında kanal dengeleme.
3.2. Dağıtılmış Dengeli Yönlendirme (DBR)
Dağıtılmış Dengeli Yönlendirme'yi (DBR) verimli bir yönlendirme protokolü olarak öneriyoruz.
zincir dışı durum kanalı ağında değer aktarımları. DBR algoritması esinlenmiştir
başlangıçta kablosuz iletişimde kullanılan BackPressure yönlendirme algoritması [7, 15] tarafından
ağlar. Gelenekselden tamamen farklı bir tasarım felsefesine dayanmaktadır.
en kısa yol yönlendirme. Özellikle, DBR herhangi bir açık yol com-
kaynaktan hedefe koyma. Bunun yerine, yönlendirme yönü,
mevcut ağın tıkanıklık gradyanları. Bir tepenin tepesinden akan suyu düşünün
tepenin eteğinde bir hedefe. Suyun rotayı bilmesine gerek yoktur.
varış yeri; tüm yapması gereken yerçekimi yönünü takip etmektir.

Sayfa 25

DBR algoritması benzer bir tasarım felsefesi kullanır ancak aynı zamanda
durum kanalı ağlarında durum bilgisi olan bağlantı modeli. Özellikle, DBR algoritması
şeffaf bir şekilde dengeyi koruyan bir durum kanalı dengeleme yeteneği ile artırılmıştır.
her durum kanalı için anced transfer akışları. Mevcut yönlendirme algoritmaları ile karşılaştırıldığında,
önerilen DBR algoritması aşağıdaki avantajlara sahiptir:
• Muhtemelen optimum verim. Başka bir deyişle, belirli bir varış oranı için
değer aktarım istekleri, eğer "destekleyen" herhangi bir yönlendirme algoritması varsa
oran, DBR de bunu yapabiliyor. "Destek" kelimesinin anlamı şurada belirtilecektir:
Bölüm 3.2.3.
• Şeffaf kanal dengeleme. DBR'de kanal yeniden dengeleme süreci
herhangi bir ek koordinatör olmadan doğal olarak yönlendirme sürecine yerleştirilmiştir.
yon. Dengeli değeri korumak için her durum kanalını otomatik olarak yeniden dengeler
uzun vadede transferler.
• Tamamen merkezi olmayan. DBR algoritması tamamen merkezi olmayan bir algoritmadır
her düğümün yalnızca komşularıyla devlet kanalı ağında konuşması gerektiği yerde
topoloji. DBR ayrıca protokolde düşük mesajlaşma maliyetine sahiptir.
• Başarısızlık direnci. DBR algoritması, arızalara karşı oldukça sağlamdır:
maksimum desteği destekleyerek yanıt vermeyen düğümleri hızlı bir şekilde algılayabilir ve
uyarlayabilir
kalan kullanılabilir düğümler üzerinden olası verim.
• Mahremiyetin korunması. Çok yollu yapısı nedeniyle, DBR algoritması nat-
urally, aktarılan değerlerin miktarı ile ilgili gizliliği korur,
ek gizlilik koruma tekniklerini kullanarak (örneğin, ZKSNARK). Daha
daha da önemlisi, DBR algoritması onion router ile sorunsuz bir şekilde entegre edilebilir.
kaynaklar / varış yerleri için anonimliğin korunması [11].
Aşağıda, önce devlet kanalı ağ modelini tanıtıyoruz, sonra açıklıyoruz
DBR algoritması ve nihayet DBR'nin performansını kanıtlayın. Unutmayın ki
açıklama kolaylığı, bu konuda iki taraflı ödeme kanallarına dikkatimizi sınırlıyoruz
bölümünde, ancak aynı fikirler değer aktarımı olan herhangi bir eyalet kanalı ağı için geçerlidir
Gereksinimler.
3.2.1. Sistem Modeli
Modelimizde, zaman, her birinin uzunluğunun
aralık genellikle bir sekmedeki fiziksel iletim gecikmesine karşılık gelir. Farz et ki

---

Sayfa 26

ağda N düğüm var. Her i ve j düğüm çifti için bir çift yönlendirilmiş
bağlantılar i → j ve j → i arasında çift yönlü bir ödeme kanalı varsa çıkılır i ↔ j
düğüm i ve düğüm j. C ij (t), t yuvasındaki i → j bağlantısının kapasitesi olsun.
ödeme kanalında düğümden transfer edilebilecek kalan bakiyeye

i bu yuvanın başındaki j düğümüne. Her biri için toplam para yatırma kısıtlaması vardır.
i düğümü ve j düğümü arasındaki iki yönlü ödeme kanalı:
c ij (t) + c ji (t) = B i↔j (t),
B i↔j (t), iki yönlü ödeme kanalı i ↔ j'nin
t slotunun başlangıcı. Toplam para yatırma B i↔j (t) 'nin zamanla değişebileceğini unutmayın.
dinamik zincir üstü fon yatırma / çekme.
Her t slotunda, her bir düğüm i dışından yeni ödeme talepleri alır.
ağ, burada düğüm k'ye teslim edilmesi gereken toplam simge miktarı
a
(k)
ben
(t) ≥ 0. Ayrıca μ ile gösterilir
(k)
ij
(t) token miktarı (teslim edilmesi gereken
düğüm k), yönlendirme değişkeni olarak adlandırılan, yuva t'deki i → j bağlantısı üzerinden gönderilir.
3.2.2. Protokol Açıklaması
DBR'nin tanımından önce, ilk olarak birkaç önemli kavramı tanıtacağız: borç
kuyruk, kanal dengesizliği ve tıkanıklık artı dengesizlik (CPI) ağırlığı.
(Borç Sırası). DBR'nin işleyişinde, her düğümün bir "borç
kuyruk uzunluğu Q olan her k düğümüne yönelik ödemeler için "queue"
(k)
ben
(t) karşılık gelir
i düğümü tarafından şu adrese aktarılması gereken belirteçlerin miktarına (hedef k ile)
bir sonraki atlama, ancak henüz t yuvasının başlangıcında aktarılmadı. Sezgisel olarak,
borç kuyruğunun uzunluğu, her bağlantıdaki tıkanıklığın bir göstergesidir. Sıra uzunluğu
evrim aşağıdaki gibidir:
Q
(k)
ben
(t + 1) =
[
Q
(k)
ben
[t) + a
(k)
ben
(t) +
∑
j∈N ben
μ
(k)
ji
(t) -
∑
j∈N ben
μ
(k)
ij
(t)
] +
,
(6)

burada [x] + = maks {0, x} (çünkü sıra uzunluğu negatif olamaz) ve N i ,
düğümün komşu düğümleri i. Yukarıdaki denklem basitçe, t yuvasındaki değişimin
kuyruk uzunluğunun nedeni üç bileşenden kaynaklanmaktadır: (1)
ağın dışında (yani, bir
(k)
ben
(t)), (2) komşulardan düğüm i'ye yönlendirilen belirteçler, yani,
25

Sayfa 27

∑
j∈N ben
μ
(k)
ji
(t) ve (3) simgeleri i düğümünden komşularına yönlendirilir, yani ie
j∈N ben
μ
(k)
ij
(t).
Hedef düğümdeki kuyruk uzunluğunun her zaman sıfır olduğu unutulmamalıdır, yani,
Q
(ben)
ben
(t) = 0, her bir i düğümü için, her paketin sonunda olabileceğini garanti eder.
DBR algoritması altında hedefine teslim edilir.
(Kanal Dengesizliği). Her i → j bağlantısı için kanal dengesizliğini şu şekilde tanımlarız:
Δ ij (t) =
∑
τ <t
∑
k
(
μ
(k)
ji
(τ) - μ
(k)
ij
(τ)
)
.
(7)
Sezgisel olarak, Δ ij (t), tarafından alınan toplam token miktarı arasındaki farktır.
j düğümünden i düğümü ve ödemeleri üzerinden i'den j'ye gönderilen toplam token miktarı
t yuvasının başlangıcına kadar kanal. Δ ij (t) <0 ise düğümün
j düğümüne j düğümünden alınandan daha fazla simge gönderdim. Açıkça, Δ ij (t)
düğüm i tarafından algılanan kanal dengesizliğinin doğal bir ölçüsü. DBR algoritmamız
ödeme kanalını , her ödeme için lim t → ∞ Δ ij (t) / t = 0 olacak şekilde dengelemeye çalışır
i ↔ j kanalı, i'den j'ye uzun vadeli gönderme hızının,
oranı j'den i'ye gönderme.
(Tıkanıklık-Artı-Dengesizlik (CPI) Ağırlığı). Tıkanıklığı-Artı- tanımlayın
İ → j bağlantısı ve hedef k için dengesizlik (CPI) ağırlığı

W
(k)
ij
(t) = Q
(k)
ben
(t) - Q
(k)
j
(t) + βΔ ij (t),
(8)
burada β> 0, kanal dengelemenin önemini ayarlayan bir parametredir. Sezgisel olarak,
yukarıdaki ağırlık, diferansiyel birikim Q'nun toplamıdır
(k)
ben
(t) - Q
(k)
j
(t) ödemeler için
i düğümü ve j düğümü (yani tıkanıklık gradyanı) arasındaki k düğümüne yöneliktir ve
i düğümü ve j düğümü arasındaki kanal dengesizliği Δ ij (t). İlki azaltmak için kullanılır
ağ tıkanıklığı ve ikincisi dengelemek için kullanılırken ağ verimini iyileştirir
ödeme kanalları.
Dağıtılmış Dengeli Yönlendirme (DBR)
Aşağıdaki protokol, her düğüm i tarafından yerel olarak yürütülür.
Her zaman diliminde t, düğüm i önce kuyruk uzunluğu bilgisini

---

Sayfa 28

komşuları ve CPI ağırlıklarını hesaplar. Sonra her bağlantı i → j, düğüm i
bu bağlantı üzerinden iletilecek en iyi ödeme akışını hesaplar:
k
*
= arg max
k
W
(k)
ij
(t).
(9)
Eğer W
(k ∗)
ij
(t)> 0, sonra μ
(k ∗)
ij
(t) = c ij (t) aksi takdirde μ
(k ∗)
ij
(t) = 0. Herhangi bir k = k ∗ için ,
μ
(k)
ij
(t) = 0.

Açıklama. Her t slotunda, DBR esasen aşağıdaki ağırlıklı toplamı çözmeye çalışır
optimizasyon sorunu:
max
∑
ij
∑
k
μ
(k)
ij
(t) W
(k)
ij
(t)
st
∑
k
μ
(k)
ij
(t) +
∑
k
μ
(k)
ji
(t) ≤ B i↔j (t), ∀i, j.
(10)
Yukarıdaki optimizasyon problemine MaxWeight da denir ve teo-
DBR'nin retikal analizi (sonraki bölüme bakın). Yukarıda bahsedilen algoritma açıklaması
MaxWeight için yaklaşık bir çözüm verir.
3.2.3. DBR'nin Verimlilik Performansı
DBR'nin çıktı performansını analiz etmek için önce birkaç tanım sunuyoruz.
• Durum kanalı ağının, eğer
lim
t → ∞
Q
(k)
ben
(t)
t
= 0, ∀i, k,
bu, her bir borç kuyruğuna uzun vadeli varış oranının,
bu kuyruktan uzun vadeli ayrılma oranı.
• Eyalet kanalı ağının, eğer
lim
t → ∞
Δ ij (t)
t
= 0, ∀ kanal i ↔ j.

Sayfa 29

Diğer bir deyişle, her bir ödeme kanalı için uzun vadeli gönderim oranını i ↔ j

i düğümünden j düğümüne, j düğümünden i düğümüne gönderme hızına eşittir.
• Tanımlamak
λ
(k)
ben
lim
t → ∞
1
t
t − 1
∑
τ = 0
a
(k)
ben
(τ)
varış yeri ile ödemeler için i düğümüne uzun vadeli ortalama varış oranı olarak
k. Varış hızı vektörü λ = (λ
(k)
ben
) i, k varsa desteklenebilir olduğu söylenir
bunun altında ağı kararlı ve dengeli tutabilen bir yönlendirme algoritması
varış hızı vektörü.
• Bir eyalet kanalı ağının aktarım hızı bölgesi, desteklenebilirler kümesidir
varış hızı vektörleri.
• Yönlendirme algoritması, herhangi bir ödemeyi destekleyebiliyorsa verim açısından optimaldir
verim bölgesi içindeki varış oranı vektörü.
Açıklama kolaylığı için, harici ödeme varış sürecinin
{a
(k)
ben
(t)} t≥0 durağandır ve kararlı durum dağılımına sahiptir ve toplam de-
her ödeme kanalı için pozitif sabit kalır, yani herhangi bir t ≥ 0 için B i↔j (t) = B i↔j
analiz, varış sürecinin durağan olmadığı durumda genişletilebilir ve
kanal para yatırma işlemleri zamana göre değişir (örneğin, dinamik zincir üzerinde para yatırma / çekme),
hantal gösterimlerin masrafı. Aşağıdaki teorem, üretim performansını gösterir.
DBR'nin mance.
Teorem 3.1. DBR algoritması verim açısından optimaldir.
Başka bir deyişle, ödemeyi saklayabilecek bir yönlendirme algoritması olduğu sürece
ağ kararlı ve dengeli, DBR algoritması da bunu başarabilir. Geri kalan
Aşağıdaki §3.2.3, Teorem 3.1'in kanıtıdır. İlk önce bir lemma tanıtıyoruz.
bir durum kanalı ağı için iş hacmi bölgesini karakterize eder.
Lemma 3.2. Varış hızı vektörü λ = (λ
(k)
ben
) i, k ancak ve ancak varsa desteklenebilir

Sayfa 30

akış değişkenleri var f = (f
(k)
ij
) i, j, k aşağıdaki koşulları sağlayan:
λ

(k)
ben
+
∑
j∈N ben
f
(k)
ji
- ∑
j∈N ben
f
(k)
ij
≤ 0, ∀k, i = k
(11)
∑
k
f
(k)
ij
=
∑
k
f
(k)
ji
, ∀i, j
(12)
∑
k
f
(k)
ij
+
∑
k
f
(k)
ji
≤ B i↔j , ∀i, j.
(13)
Kanıt. Yukarıdaki koşulların gerekliliği önemsizdir. Eşitsizlik (11) karşılık gelir
akış koruma gereksinimine. İhlal edilirse, i düğümüne varış hızı
kalkış oranından daha büyük ve eyalet kanal ağı kararsız. Denklem
(12) kanal bakiyesi gerekliliğine karşılık gelir. İhlal edilirse, o zaman kanal
i ↔ j dengesizdir. Eşitsizlik (13), kanal kapasitesi kısıtlamasına karşılık gelir,
i ↔ j her kanal üzerinden aktarılan jetonların toplamı toplamı geçemez.
kanal para yatırma B i↔j .
Yukarıdaki koşulların yeterliliğini kanıtlamak için bir algoritma oluşturuyoruz
bu, varış hızı vektörü olduğunda durum kanalı ağını stabilize edebilir ve dengeleyebilir
λ, (11) - (13) 'ü karşılar. Algoritma basittir: her t yuvasında yönlendirmeyi ayarlayın
değişken μ
(k)
ij
(t) = f

(k)
ij
herhangi bir i için, j, k. Açıkça, bu yönlendirme algoritması altında her kanalda
nel i ↔ j her t yuvasında dengeli kalır, çünkü ∑
k
μ
(k)
ij
(t) = ∑
k
f
(k)
ij
= ∑
k
f
(k)
ji
=
∑
k
μ
(k)
ji
(t). Dahası, ağ, akış kontrolünden dolayı algoritma altında kararlıdır.
her düğüm için hizmet gereksinimi karşılanır. Her bir i → j bağlantısının sahip olabileceğine dikkat edin.
başlangıçta yetersiz fon (yani, c ij (0) <∑
k
f
(k)
ij
) öyle ki yönlendirme kararı,
mümkün. Bu durumda, j düğümünün başlangıçta bazı jetonları i düğümüne aktarmasına izin verebiliriz.
devlet kanalının her iki ucundaki fonu eşitlemek için. Böyle bir ayarlama
süreç , her i ↔ j kanalı için en fazla B i↔j sub-optimal transferlere maruz kalır ve
uzun vadede ağ istikrarını ve kanal dengesini etkilemez.
Bu nedenle, denklemler (11) - (13) bir varış için gerekli ve yeterli bir koşuldur.
hız vektörü λ desteklenebilir.
Lemma 3.2 ispatında belirtilen yönlendirme algoritmasının
dış varış hızı vektörünü bilmediğimiz için pratikte uygulanamaz
λ önceden. Aşağıda, DBR'nin aynı verimi elde edebileceğini kanıtlıyoruz
önceden herhangi bir ödeme trafiği istatistiğini bilmeden performans.

Sayfa 31

Lemma 3.2'ye göre, bir varış hızı vektörü λ, iş hacmi bölgesine aitse,
(11) - (13) 'ü karşılar ve ispatında belirtilen algoritma tarafından desteklenebilir
Lemma 3.2 (optimal oracle algoritması olarak anılır). Aşağıda, şununla belirtin:
˜μ
(k)
ij
(t) t yuvasındaki optimal oracle algoritması tarafından verilen yönlendirme kararı. Tarafından
optimal oracle algoritmasının doğası, bizde ˜μ

(k)
ij
(t) = f
(k)
ij
herhangi bir t için (görmezden geliyorsa
ilk fon ayarlama süreci).
Lyapunov işlevini aşağıdaki gibi tanımlayın:
Φ (t) =
∑
ben, k
(
Q
(k)
ben
(t)
) 2
+
β
2
∑
ben, j
Δ 2
ij (t).
(14)
Ayrıca koşullu Lyapunov kaymasını D (t) olarak tanımlayın
E [Φ (t + 1) - Φ (t) | Q (t), Δ (t)],
Beklentinin gelişlerin rastlantısallığı ile ilgili olduğu yerde. Kolaylaştırmak için
analiz, her birinde ağa gelen yeni ödemelerin miktarının
yuva bazı sabitlerle sınırlıdır. Denklem (6) ile, elimizde
(
Q
(k)
ben
(t + 1)
) 2
=
[
Q
(k)
ben
[t) + a
(k)
ben
(t) +
∑
j
μ
(k)
ji
(t) -
∑
j
μ
(k)

$$
\begin{aligned}
&ij \\
&(t) \\
&] 2 \\
&= \\
&(\\
&Q \\
&(k) \\
&ben \\
&(t) + \\
&\sum \\
&j \\
&\mu \\
&(k) \\
&ji \\
&(t) - \\
&\sum \\
&j \\
&\mu \\
&(k) \\
&ij \\
&(t) \\
&) 2 \\
&+ \\
&(\\
&a \\
&(k) \\
&ben \\
&(t) \\
&) 2 \\
&+ 2a \\
&(k) \\
&ben \\
&(t) \\
&(\\
&Q \\
&(k) \\
&ben \\
&(t) + \\
&\sum \\
&j \\
&\mu \\
&(k) \\
&ji \\
&(t) - \\
&\sum \\
&j \\
&\mu \\
&(k) \\
&ij \\
&(t) \\
&) \\
&\leq \\
&(\\
&Q \\
&(k)
\end{aligned}
$$

ben
(t)
) 2
- 2Ç
(k)
ben
(t)
(∑
j
µ
(k)
ij
(t) -
∑
j
µ
(k)
ji
(t)
)
+ 2a
(k)
ben
(t) Q
(k)
ben
[t) + sabit,
(15)
eşitsizliğin, gelişin bir
(k)
ben
(t) her yuvada t
bazı sabitler ve her birinde aktarılan jeton sayısının
yuva da sınırlıdır (çünkü µ
(k)
ij
(t) ≤ B i↔j). Şimdi sahibiz
∑
ben, k
(
Q
(k)
ben
(t + 1)
) 2
- ∑
ben, k
(
Q
(k)
ben
(t)
) 2
≤ sabit - 2∑
ben, k

Q
(k)
ben
(t)
(∑
j
μ
(k)
ij
(t) -
∑
j
μ
(k)
ji
(t)
)
+ 2
∑
ben, k
a
(k)
ben
(t) Q
(k)
ben
(t)
= sabit - 2
∑
ben, j
∑
k
μ
(k)
ij
(t)
(
Q
(k)
ben
(t) - Q
(k)
j
(t)
)
+ 2
∑
ben, k
a
(k)
ben
(t) Q
(k)
ben
(t),

Sayfa 32

yukarıdaki eşitlikte toplamı yeniden düzenlediğimiz yer. Benzer şekilde, bunu fark etmek
$\Delta_{ij}(t+1) = \Delta_{ij}(t) + \sum_k \mu_{ji}^{(k)}(t) - \sum_k \mu_{ij}^{(k)}(t)$,
kanıtlayabiliriz
$\beta_2 \sum_{ben,j} \Delta_{ij}^2(t+1) - \beta_2 \sum_{ben,j} \Delta_{ij}^2(t) \le \beta \cdot sabit - \beta \sum_{ben,j} \sum_k \Delta_{ij}(t) \left(\mu_{ij}^{(k)}(t) - \mu_{ji}^{(t)} \right) = \beta \cdot sabit - \beta \sum_{ben,j} \sum_k \mu_{ij}^{(t)} \left(\Delta_{ij}(t) - \Delta_{ji}(t) \right)$

)
= β · sabit - 2
∑
ben, j
∑
k
μ
(k)
ij
(t) βΔ ij (t),
,
(16)
Δ ij (t) = −Δ ji (t) olduğu gerçeğini kullandığımız yerde . Sonuç olarak, (15) ve (16) birleştirilerek, koşullu Lyapunov kayması aşağıdakilerle sınırlandırılabilir:
D (t) ≤ βc 1 + c 2 - 2∑
ben, j
∑
k
μ
(k)
ij
(t)
(
Q
(k)
ben
(t) - Q
(k)
j
(t) + βΔ ij (t)
)
+ 2
∑
ben, k
λ
(k)
ben
Q
(k)
ben
(t)
≤ βc 1 + c 2 - 2∑
ben, j
∑
k
˜μ
(k)
ij
(t)
(
Q
(k)
ben
(t) - Q
(k)

j
(t) + βΔ ij (t)
)
+ 2
∑
ben, k
λ
(k)
ben
Q
(k)
ben
(t)
= βc 1 + c 2 - 2∑
ben, j
∑
k
f
(k)
ij
(
Q
(k)
ben
(t) - Q
(k)
j
(t) + βΔ ij (t)
)
+ 2
∑
ben, k
λ
(k)
ben
Q
(k)
ben
(t)
= βc 1 + c 2 + 2
∑
ben, k
Q
(k)
ben
(t)
(
λ
(k)
ben
+
∑
j
f
(k)

ji
- ∑
j
f
(k)
ij
)
- β ∑
ij
∑
k
Δ ij (t)
(
f
(k)
ij
- f
(k)
ji
)
≤ βc 1 + c 2 ,
c 1 ve c 2 bazı sabitler ise, ikinci eşitsizlik işlemden kaynaklanmaktadır
DBR (bkz. (10)) ve son eşitsizlik (11) ve (12) 'den kaynaklanmaktadır. Yasasını kullanmak
yinelenen beklentiler şunları sağlar:
E [Φ (τ + 1)] - E [Φ (τ)] ≤ βc 1 + c 2 .
T = 0, ⋯, t - 1 üzerinden toplanırsak, elimizde
E [Φ (t)] - E [Φ (0)] ≤ (βc 1 + c 2) · t.

Sayfa 33

O zaman bizde
∑
ben, k
E
[(
Q
(k)
ben
(t)
) 2]
+
β
2
∑
ben, j
E
[
Δ 2
ij (t)
]
≤ (βc 1 + c 2) t + E [Φ (0)].
(17)
DBR'nin kanal bakiyesine ulaştığını göstermek için (17] 'den şunu not ediyoruz:
β
2

(
E [
∑
ben, j
| Δ ij (t) |]
) 2
≤
β
2
∑
ben, j
E
[
Δ 2
ij (t)
]
≤ (βc 1 + c 2) t + E [Φ (0)],
ilk eşitsizliğin geçerli olduğu yerde, çünkü | Δ (t) | negatif olamaz, yani
Var (| Δ (t) |) = E
[∑
ben, j
Δ 2
ij (t)
]
-
(
E [
∑
ben, j
| Δ ij (t) |]
) 2
≥ 0. Böylece elimizde
E [
∑
ben, j
| Δ ij (t) |] ≤
√
2c 1 t +
2c 2 t
β
+
2E [Φ (0)]
β
.
E [Φ (0)] <∞ olduğundan, herhangi bir ödeme kanalı için buna sahibiz i ↔ j
lim
t → ∞
E
[| Δ ij (t) |
t
]
= 0,
yani ağ, DBR algoritması altında kanal dengesini korur.
Benzer şekilde bunu gösterebiliriz
∑

ben, k
E [Q
(k)
ben
(t)] ≤
√
(βc 1 + c 2) t + E [Φ (0)],
ki bunu ima eder
lim
t → ∞
E [Q
(k)
ben
(t)]
t
= 0, ∀i, k,
yani ağ, DBR algoritması altında kararlıdır.
32

Sayfa 34

3.3. DBR Tartışmaları
3.3.1. Başarısızlık Direnci
Uyarlanabilir ve çok yollu yapısı nedeniyle, DBR algoritması doğası gereği sağlamdır
ağ arızalarına karşı. Örneğin, yanıt vermeyen düğümler olduğunda DBR,
kalan kullanılabilirlik üzerinden mümkün olan maksimum verimi hızla uyarlayın ve destekleyin-
mümkün düğümler.
3.3.2. Gizlilik
DBR'nin çok yollu doğası nedeniyle, herhangi bir ara düğüm yalnızca
her bir değer transferi talebinin küçük bir kısmı için bilgi. Sonuç olarak, DBR
algoritması, doğal olarak, işlem miktarı açısından iyi bir gizlilik koruması sağlar.
ferred değerler. Bununla birlikte, DBR'de her düğümün her birinin hedefini bilmesi gerekir.
uygun bir borç kuyruğuna yerleştirmek için değer transferi talebi. Eğer biz de saklamamız gerekiyorsa
ödeme hedefi, soğan yolu [11] DBR ile bağlantılı olarak kullanılabilir. İçinde
onion yönlendirme, mesajlar şifreleme katmanları içinde kapsüllenir. Şifrelenmiş veriler
her biri soğan düğümleri adı verilen bir dizi ağ düğümü aracılığıyla iletilir.
Tek bir katmanı "soyarak" verilerin bir sonraki hedefini ortaya çıkarır. Final ne zaman
katmanın şifresi çözülür, mesaj hedefine ulaşır. Ödemeleri yönlendirebiliriz
soğan düğümlerinden oluşan bir overlay ağı aracılığıyla ve en uygun şekilde DBR'yi uygulayın
soğan düğümleri arasında rota ödemeleri.
3.4. Simulasyon sonuçları
DBR'nin performansını mevcut iki yönlendirme algo ile sayısal olarak karşılaştırıyoruz.
ödeme durumu kanal ağlarında rithms: SpeedyMurmurs [13] ve Flare [10] (kullanılmış
Yıldırım Ağında). Simülasyon, Şekil 2'de verilen topoloji üzerinde gerçekleştirilir.
3.
Şekil 4, çıktı performans karşılaştırmasını göstermektedir. DBR'nin
Mevcut ödeme hacmine kıyasla ortalama ödeme hacminde 15 kat iyileşme sağlar
yönlendirme algoritmaları. DBR'nin olağanüstü verim performansı,
kanal dengeleme ve tıkanıklığa duyarlı doğa. Özellikle, Şekil 5,
gözlemleyebileceğimiz DBR, SpeedyMurmurs ve Flare altında kanal kullanımı 3
3 Kanal kullanımı, her bir zaman diliminde aktarılan jeton miktarı arasındaki orana karşılık gelir ve
33

Sayfa 35

rce Yönlü Grafik

Şekil 3. Simülasyonlarda kullanılan ödeme kanalı ağ topolojisi (77 düğüm, 254 çift yönlü ödeme kanalları). Ödeme kanalı ağı, 40 ödeme akışı ile çalışıyor.
rastgele seçilen kaynak-hedef çiftleri. Her kanal için ilk depozito eşit şekilde
[100,200] jeton içinde dağıtılır. Ödeme varışları bir Poisson sürecini takip eder ve
her ödeme, ortalama 3 jetonla geometrik bir dağılımı takip eder.

Şekil 4. Anında Ödeme işleme hızı
DBR arasında karşılaştırma (ort: 6748 ödeme-
ments / slot), SpeedyMurmurs (ort: 467
ödemeler / yuva) ve Flare (ortalama: 316 ödeme-
yönlendirmeler / yuva).

Şekil 5. Kanal kullanım karşılaştırması
DBR, SpeedyMurmurs ve Flare arasında.
Daha yüksek kanal kullanımı, daha yüksek bir
kanal dengeleme seviyesi.
DBR'nin sürekli olarak yüksek (neredeyse% 100) kanal kullanımına ulaştığını, diğerinin ise yönlendirme algoritmaları, eksikliğinden dolayı yalnızca% 5'ten daha az kanal kullanımı elde eder kanal dengeleme.
tüm kanalların toplam token depozitoları. Örneğin, toplam depozito miktarı 100 ve sadece 50 ise t yuvasına taşınırsa, bu yuvadaki toplam kanal kullanımı% 50'dir.

Sayfa 36

4. cOS: Zincir Dışı Merkezi Olmayan Uygulama İşletim Sistemi
Herkesin hızla ölçeklenebilir zincir dışı merkezi olmayan uygulamaları oluşturmasına, çalıştırmasına ve kullanmasına yardımcı olmak için
zincir dışı kaynaklı ek karmaşıklıklarla uğraşmadan katyonlar
Celer Network, daha yüksek bir soyutlama düzeyinde yenilikler yapıyor: cOS,
uygulama geliştirme çerçevesi (SDK) ve çalışma zamanı sistemi. Bu bölüm şunları sağlar:
üst düzey vizyon, tasarım hedefleri ve cOS üzerindeki resimler.
4.1. Koşullu Bağımlı Durumların Yönlendirilmiş Döngüsel Grafiği
Bu bölümde, soyutlama modelimizin zincir dışı yapıya ilişkin bir görünümünü sunuyoruz.
uygulamalar ve modelin durum kanalı ağlarıyla nasıl bütünleştirildiğini açıklayın.
Basit P2P ödemelerinin ötesinde kullanım örneklerini desteklemek için bir sistem modelliyoruz:
Koşullu olarak bağımlı bir yönlendirilmiş döngüsel olmayan grafik (DAG) olarak zincir dışı uygulamalar
kenarların aralarındaki bağımlılıkları temsil ettiği durumlar.
Şekil 6. Koşullu Bağımlı Durumların DAG'si
Şekil 6, Genelleştirilmiş Koşullu Ödemenin
Ödeme ağlarındaki kanal, zincir üzerindeki devletlerle yapılan tek sözleşmedir. The
Bu zincir üzerindeki durumların çözümü, bir veya daha fazla koşullu ödemeye bağlıdır.
Tamamen zincir dışı ancak zincir içi olan jekler (ör. Koşullu Ödeme Nesnesi 3)
uygulanabilir. Bu koşullu ödeme nesnelerinin yalnızca
basit zaman karması kilitli işlemler, ancak zincir dışı uygulamaya koşullandırılabilir

---

Sayfa 37

Şekil 6'daki "Zincir Dışı Uygulama X" gibi sözleşme durumları.
Koşullu ödeme nesneleri, tıpkı
basit koşulsuz ödeme nesneleri. Örneğin, 1. Ödeme Kanalı bir
Alice ve Bob'u bağlayan kanal ve Payment Channel 2 birbirine bağlanan bir kanal olabilir
Bob ve Carl. "Zincir dışı Uygulama 2", Alice'in oynadığı zincir dışı bir satranç oyunu olsun
Carl, ve Alice'in "Alice, Carl'a 10 ETH ödeyecek
Carl oyunu kazanırsa ”. Alice ve Carl arasında doğrudan bir kanal olmasa bile, Alice
Ali'ye Bob aracılığıyla iki koşullu koşullu ödeme gönderebilir
kilitler. İlk katman, Bob'un geçiş yapmasını sağlamak için basit bir zaman karma kilididir ve
ödemeyi makul bir sürede çözer. İkinci katman,
satranç oyununun sonucuna göre ödeme koşulu. Bu iki atlamalı röle ile,
Alice ile Carl arasındaki şartlı ödeme, Bob aracılığıyla çözülebilir.
satranç oyununa dahil olmadı. Bu, bağımlılığın nasıl küçültülmüş bir örneğidir.
genelleştirilmiş durum kanalları, koşullu ödeme nesneleri ve zincir dışı tarafından oluşturulan grafik
uygulamalar, rastgele karmaşık çok partili etkileşimleri destekleyebilir.
Zincir dışı nesnelerin yalnızca zincir dışı nesnelere bağlı olması gerekmediğini unutmayın. İçin
Örneğin Alice, Carl'a belirli bir ENS'yi başarıyla aktardığında ödeme yapabilir.
eski adı. Başka bir deyişle, ödeme zincir dışı duruma bağlıdır
ENS adının sahibinin Carl'dan Alice'e değiştiği.
Ayrıca, zincir dışı ödeme nesnelerinin her zaman koşullu olması gerekmez: bir koşul
tional ödeme nesnesi, koşulsuz bir denge kanıtı olarak "yozlaşabilir".
uygulama çalışır. Daha genel olarak konuşursak, koşullu bağımlılıklar geçicidir
doğa: uygulama durumu güncellemeleri, bir çift topolojik geçiş yoluyla yapılır.
temel durum grafiği. İlk geçiş ileri yönde gider ve
ikincisi ters yönde. Zincir üzerinden başlayarak ileri çapraz geçiş
durum kanalı sözleşmeleri, ek geçici koşullu bağımlılık kenarları oluşturur ve
mevcut olanları değiştirir. Ters geçiş, mevcut geçici koşullu durumu kaldırabilir
bağımlılık kenarları, çünkü bazı koşullar geçerken sabit doğru olarak değerlendirilir
geriye.
4.2. Zincir dışı Uygulama Geliştirme Çerçevesi
Modern yüksek seviyeli dillerin ve işletim sistemlerinin

temel donanım hakkında ayrıntılar, koşullu ile etkileşimin karmaşıklığı
36

Sayfa 38

durum bağımlılık grafikleri, özel bir geliştirme çerçevesi gerektirir. İle
“kullanım kolaylığı” ilkesiyle Celer Network, bir bilgisayar olan cOS SDK'yı sunar.
zincir dışı durumların oluşturulması, izlenmesi ve çözümlenmesi için plete araç zinciri çözümü
uygulamalar. SDK'nın zincir dışı ölçeklendirmenin benimsenmesini hızlandıracağını umuyoruz.
çözüm ve ödeme ağı, Celer Network tarafından sağlanan, güçlü bir
ekosistem.
cOS API
Akıllı sözleşme
Platforma özel kod
Durum bağımlılık grafiği
cOS durum derleyicisi
cApp
Şekil 7. Celer Network'teki (cApp) merkezi olmayan bir uygulamanın yapısı
Genel olarak, merkezi olmayan uygulamaları iki sınıfa ayırıyoruz: basit ödeme
kullanım başına uygulamalar ve daha karmaşık çok partili uygulamalar. Kullanım başına ödeme
uygulaması
öneriler, kullanıcının almaya devam ettiği Orchid Protokolü gibi örnekleri içerir.
gerçek dünya varlığından mikro hizmetler (ör. veri aktarımı) ve ödemeleri
ödeme ağı. Diğer kapalı alanlara koşullu bağımlılığa gerek olmadığından
zincir durumları, yönlendirme katmanının üstünde bir yalın taşıma katmanı API'si, her ikisi de
Celer Network tarafından sağlanan bu tür durumlar için yeterlidir.
Genel yapısı gösterilen çok partili uygulamalar sınıfı
Şekil 7'de, koşullu durum bağımlılığı grafikleri fikrinin gerçekten parladığı yerdir.
SDK, geliştiricilerin kullanması için bir dizi tasarım modeli ve ortak bir çerçeve tanımlar.
Koşullu bağımlılıkları ifade eder. Mevcut akıllı sözleşmeyi uzatmayı planlıyoruz
metaprogramlama gibi modern yazılım yapım tekniklerine sahip diller,
ek açıklama işleme ve bağımlılık ekleme, böylece bağımlılık bilgileri
çok müdahaleci olmadan açıkça yazılabilir. Bir derleyici daha sonra işler
uygulama kodu, bildirilen zincir dışı nesneleri çıkarır ve
ikili bağımlılık grafiği. Derleyici, geçersiz veya yerine getirilemeyen bir bağımlılık tespit etti
geliştiricinin hata ayıklamasına yardımcı olmak için bilgi verir ve insan tarafından okunabilir hatalar
oluşturur.
Geliştiricilerin bağımlılıklar hakkında daha fazla akıl yürütmesine yardımcı olmak için SDK,
grafikleri Graphviz gibi ortak formatlara serileştirmek için
37

Sayfa 39

kolayca görselleştirilebilir ve sunulabilir.
SDK ayrıca bir dizi "köprü yöntemi" oluşturan bir kod oluşturucu sağlar
derleme zamanında kodu mevcut olan akıllı sözleşmelerle etkileşim için. The
kod üreteci, uygulama ikili arabirimini (ABI) ayrıştırır,
akıllı bir sözleşmedeki tüm çağrılabilir işlevlerin doğası ve karşılık gelen
Java gibi platforma özgü dillerde köprü yöntemleri. Ana avantajı
bu yaklaşım tür güvenliğidir: tutkal yöntemleri, yöntem imzalarını kopyalar.
Statik ve sağlam bir derleme zamanı sağlayan akıllı sözleşmedeki işlevleri sadık bir şekilde
yöntemi yürütmek için cOS çalışma zamanına göndermeden önce kontrol edin.
4.3. Zincir dışı Uygulama Çalışma Zamanı
COS çalışma zamanı, cApps 4 ve Celer Network aktarımı arasında arayüz görevi görür.
bağlantı noktası katmanı. Hem ağ iletişimi hem de yerel off-
zincir durumu yönetimi. Genel mimari Şekil 8'de gösterilmektedir.

Akıllı sözleşme sanal makinesi
Sanal makine yerel köprü
Yerel depolama
cApp
Devlet koruyucu ağı
Blockchain
cApp
cApp
cChannel
İletişim kurmak
Kalıcı
İhtilaf
Elini
cOS
Şekil 8. cOS Çalışma Zamanı Mimarisi
Ağ cephesinde, çalışma zamanı, çok taraflı iletişimi yönetir.
cApp'ın yaşam döngüsü. Aynı zamanda güvenli çok partili bileşenler için bir dizi ilkel sağlar.
oyun oynama gibi karmaşık kullanım durumlarını destekleyebilen putation. Bu durumuda
karşı taraf arızası, ister arıza-durdurma isterse Bizans olsun, çalışma zamanı ihtilafları
zincir üzerindeki durum. İstemcinin çevrimdışı olması durumunda, çalışma zamanı mevcut-
State Guardian Ağına yük aktarımı. Müşteri tekrar çevrimiçi olduğunda,
çalışma zamanı, yerel durumları State Guardian Network ile senkronize eder.
4 Celer Network üzerinde çalışan merkezi olmayan uygulamaları cApps olarak adlandırıyoruz.

Sayfa 40

Yerel zincir dışı durum yönetimi için, koşullu durum grafikleri tarafından sentezlenen
cOS SDK, cApp içinde paketlenir ve zincir dışı kullanım için çalışma zamanına aktarılır
yürütme. Çalışma zamanı, oluşturmak, güncellemek, depolamak ve izlemek için altyapı görevi görür.
itor zincir dışı durumları Celer Network istemcilerinde yerel olarak belirtir. İç mantığını izler
üzerinde çalışan uygulamalar ve durum yukarı DAG geçişini gerçekleştirir.
§ 4.1'de özetlendiği gibi tarihler. Ayrıca, aşağıdaki gibi ödeme güvenilirliği sorunlarını incelikle ele alır.
ödemeyi yönlendirmek için yetersiz kapasite.
COS çalışma zamanı, özünde, çalıştırmak için yerel bir sanal makine (VM) paketler
akıllı sözleşmeler. COS'u masaüstü dahil birçok platforma dağıtmayı planlıyoruz.
top, web, mobil ve IoT cihazları için iddialı tasarım ilkesini benimsedik.
"bir kez yazın, her yerde çalıştırın". Diğer bir deyişle, geliştiricilerin ortak
iş mantığını bir kez kullanın ve her yerde aynı zincir üzerinde akıllı sözleşme kodunu çalıştırın
Aynı mantığın birden çok varyantını uygulamak zorunda kalmanın tersine vironment. Tarafından
Bu prensibi benimseyerek, kod tekrarını ortadan kaldırmayı ve yüksek derecede
çeşitli platformlarda tutarlılık.
CApps'in kullanıcı arabirimi (UI) gibi platforma özgü kısmı,
her platforma en uygun diller (örn. Android için Kotlin ve iOS için Swift).
UI kodu ayrıca platforma özgü yardımcı programları ve kitaplıkları kullanmakta serbesttir, böylece görünüm
ve cApps hissi, her platformdaki ilgili tasarım yönergeleriyle eşleşir.
COS çalışma zamanı, farklı dillerde VM'de yerel köprü uygulamaları sağlar
platforma özgü kodun temeldeki iş mantığı ile etkileşime girmesi için. Eski için
Örneğin, iOS üzerinde çalışan bir satranç oyununu temsil eden ve kullanıcı arayüzü yazılı olan bir cApp düşünün
Swift ve Solidity ile yazılmış iş mantığı. Doğal olarak, kullanıcı arayüzü katmanının
oyun tahtasının durumu için cOS VM'yi sorgulayın ve bunu aracılığıyla yapabilecektir.
Solidity-Swift köprüsü. Sözleşmenin kodu derlemede mevcut olduğundan
kod üreteci cOS SDK, adında bir köprü yöntemi oluşturabilirdi.

Gerçek sorgu için sanal makineye gönderilen chess.getBoardState. Her ne zaman
mümkünse, dilin yabancı işlev arayüzünü (ör. JNI) kullanırız.
akıllı sözleşmeler ve yerel kod arasında ileri geri arama ek yükü.
Geliştirici, aynı hata ayıklama ve profil oluşturma araçlarını da kullanabilecektir.
zincir dışı geliştirme senaryosunda zincir içi akıllı sözleşmeler.
Durum değişikliklerini gerçekten kopyalamak için ...
zincir dışı ortamda, sanal makine aynı bayt kodu ile ilerler.
39

Sayfa 41

birkaç farklılığın ikazıyla zincirleme olarak infaz edilselerdi. İlk büyük
fark, VM'nin durumları on yerine yerel olarak güncellemesi ve depolaması gerektiğidir.
blok zinciri. Sanal makine arasında sorunsuz ve şeffaf bir birlikte çalışma elde etmek için
ve cOS'un geri kalanı için platforma özgü köprü oluşturan bir dizi API uygulayacağız.
VM ile depolama arka uçları. İkinci büyük fark şudur:
her zaman çevrimiçi olduğundan, yerel bir VM herhangi bir zamanda beklenmedik bir şekilde
kapanabilir.
yazılım hataları, donanım arızası veya basitçe güç kaybı. Yerelin bozulmasını önlemek için
sağlam bir günlük kaydı, kontrol noktası belirleme ve işleme protokolü uygulamamız gerekiyor.
Üçüncü bir küçük fark, gaz ölçümü mantığının atlanabilmesidir, çünkü
uygulama yerel olarak gerçekleşir ve gaz ücretlerinin alınması mantıklı değildir.
Birlikte verilen sanal makinenin iyi çalışması için hafif ve performanslı olması gerekir
sıkı işlemci gücü altında çalışma eğiliminde olan mobil ve IoT cihazlarında,
kapasite ve pil ömrü kısıtlamaları. Şu anda hafif bir
COS'ta Ethereum VM, daha yaygın bytecode for-
daha fazla sözleşme dilini desteklemek amacıyla matlar (ör. WebAssembly) ve
diğer blok zincirleri.
Nihai cOS VM vizyonumuzda, aşağıdakiler gibi modern VM tekniklerini uygulayacağız:
Vaktinden önce derleme (AOT) ve tam zamanında derleme (JIT) ile neredeyse
zincir dışı akıllı sözleşme uygulamasının yerel performansı. Yorumlamak yerine
Ethereum sanal makinelerinin çoğunun şu anda yaptığı gibi akıllı sözleşme bayt kodlarını derliyoruz.
bayt kodlarını yerel koda daha yakın olan daha düşük seviyeli ara temsillere dönüştürür.
Belirli bir sözleşmenin kodu derleme zamanında mevcutsa (ör. Bir sözleşme
halihazırda zincir üzerinde konuşlandırılmış), derlemeyi önceden ve statik olarak gerçekleştiririz
ikiliyi uygulamanın geri kalanıyla ilişkilendirin. Dinamik olarak yapılan sözleşmeler için
çalışma zamanında yüklenirse, sık sık çağrılan işlevler (yani "etkin" kod) için bunların profilini
çıkarırız
ve tam zamanında derleme yapın. Bu ikisinin kombinasyonunun
teknikler, performans ve enerji tüketimi arasında büyük bir denge sağlayacak,
hem mobil hem de IoT cihazları için çok önemlidir.
5. cEconomy: Zincir Dışı Kriptoekonomi Mekanizması Tasarımı
Celer Network'ün yerel dijital kriptografik olarak güvenli protokol belirteci,
(CELR), Celer Network'teki ekosistemin önemli bir bileşenidir ve
40

Sayfa 42

yalnızca ağda kullanılmak üzere imzalanmıştır. CELR, iade edilmeyen işlevsel bir yardımcı
programdır
Celer Net'te ekosistemde platform para birimi olarak kullanılacak jeton
iş. CELR hiçbir şekilde herhangi bir pay sahipliği, katılım, hak, unvan,
veya Token Satıcısı, Vakıf, iştirakleri veya başka herhangi bir şirketteki çıkar,
girişim veya taahhüt, ne de CELR token sahiplerine herhangi bir ücret vaadinde bulunma hakkı
vermez,
gelir, kar veya yatırım getirisi ve menkul kıymet teşkil etmesi amaçlanmamıştır

Singapur'da veya ilgili herhangi bir yargı alanında. CELR yalnızca Celer'de kullanılabilir
Ağ ve CELR sahipliği, aşağıdakiler dışında açık veya zımni hiçbir hak taşımaz:
CELR'yi, Celer'ın kullanımını ve Celer ile etkileşimi sağlamak için bir araç olarak kullanma hakkı
Ağ.
Aşağıda, Celer Network'ün kriptoekonomi mekanizmalarını, cEcon-
omy, tasarımı iyi bir kriptoekonomi modeli (token
model) ek değerler sağlamalı ve yeni oyun-teorik dinamikleri etkinleştirmelidir
aksi halde imkansızdır. Aşağıda, ilk olarak temel bilgileri açıklıyoruz
zincir dışı ekosistemlerdeki ödünleşmeler (Bölüm 5.1) ve ardından cEconomy'nin
değer getirebilir ve bu değiş tokuşları "dengelemek" için yeni dinamikler etkinleştirebilir (Bölüm
5.2).

5.1. Zincir Dışı Ekosistemlerde Ödünleşimler

Herhangi bir zincir dışı çözüm, ölçeklenebilirlik kazanırken, aynı zamanda ödünler de
veriyor. Aşağıda-
ing, zincir dışı ekosistemlerde iki temel değiş tokuşu tanımlıyoruz: ölçeklenebilirlik-likidite
değiş tokuşlar ve ölçeklenebilirlik-kullanılabilirlik ödünleri.

5.1.1. Zincir Dışı Ölçeklenebilirlik ve Likidite

Zincir dışı platform, önce ağ likiditesinden vazgeçerek ölçeklenebilirlik kazanır. Örneğin,
iki taraflı bir ödeme devlet kanalında, ilgili iki taraf güvenli bir şekilde birbirlerini gönderebilir
temel blok zincirine çarpmadan yüksek hızlarda ödemeler
başlangıçta zincir içi tahvil sözleşmesine yatırılan likidite. Likidite kilitlemesi
bu doğanın çalışması son kullanıcılar için iyidir, çünkü son kullanıcılar basitçe para yatırabilirler.
açık kanallara kendi likiditesi ve ölçeklenebilir dApp'lerin tadını çıkarın. Ancak, bir
Zincir dışı Servis Sağlayıcı olarak çalışmak isteyenler için önemli zorluk
(OSP'ler). Eyalet kanallarını örnek olarak kullanarak, OSP'lerin her birinde para yatırması gerekir.
ödeme imkanı olan kanal. Bu mevduatlar kolaylıkla bir

Sayfa 43

astronomik miktar. Celer Networks yan zincir kanalları önemli ölçüde
Likidite gereksinimi düzeyini düşürürseniz, her blok teklif verenin yine de yatırması gerekir
“tehlikede” değer transferi seviyesiyle orantılı sahtekarlığa dayanıklı tahviller.
Sonuç olarak, etkin zincir dışı hizmet sağlamak için önemli miktarda likidite gereklidir.
küresel blockchain kullanıcıları için kötülükler. Ancak balinaların ticari menfaati olmayabilir
veya zincir dışı bir hizmet altyapısını çalıştırmak için teknik yeterliliğe sahipken,
güvenilir ve ölçeklenebilir bir zincir dışı hizmet çalıştırmanın teknik yeteneği genellikle
Kanal mevduatları veya sahtekarlığa dayanıklı tahviller için yeterli sermayeye sahip değilsiniz. Böyle bir uyumsuzluk
zincir dışı platformların kitlesel olarak benimsenmesi ve teknik gelişimi için büyük bir engel oluşturur.
formlar. Azaltılmazsa, sonunda yalnızca zenginler OSP'ler olarak hizmet edebilir. Bu yüksek sermaye
bir OSP olma engeli, yetersiz sağlayan merkezi bir ağ ile sonuçlanacaktır.
Blockchain'in ademi merkeziyet vizyonunun tüm öncülünü araştırıyor. Daha pratik bir
görüntüleme, sansür, düşük hizmet kalitesi ve gizlilik ihlali, günümüzde olduğu gibi kullanıcılara zarar verecektir.
merkezi hizmetler yapar.

5.1.2. Zincir Dışı Ölçeklenebilirlik ve Kullanılabilirlik

Zincir dışı bir platform, uygulama durumlarını devre dışı bırakarak ölçeklenebilirliği iyileştirirken
zincir, kullanıcılara pratik olmayan "her zaman çevrimiçi" bir sorumluluk yükler, çünkü
Zincir dışı durumlar her zaman zincir içi anlaşmazlıklar için mevcut olmalıdır. Örneğin, bi-
parti ödeme durumu kanalı, bir taraf çevrimdışı olursa karşı taraf saldırıya uğrayabilir
ya da kötü niyetli davranır ve kendisi için eski ama daha elverişli bir duruma yerleşmeye çalışır. The
veri kullanılabilirliği sorunu, blok önericilerinin bulunduğu bir yan zincir kanalında daha da kritiktir.
katılımcılar çevrimdışıyken bağımsız olarak izlenmeli ve doğrulanmalıdır;
bu bir güvenlik meselesidir ve dikkatle incelenmelidir. Bu meydan okuma bile
IoT cihazlarının bulunduğu makineden makineye iletişim senaryolarında daha kritik

her zaman çevrimiçi olma olasılığı düşüktür. Bu nedenle, uygun mekanizmaları tasarlamak çok önemlidir
zincir dışı bir platformda veri kullanılabilirliğini garanti eden. Bu sorunu çözmek için
tüm zincir dışı ekosistemin sistematik düşünmesi ve mevcut çözümlerin tümü başarısız
ademi merkeziyetçilik, verimlilik, basitlik, esnekliğin önemli özelliklerini sağlamak,
ve aşağıdaki bölümde daha fazla tartışacağımız gibi güvenlik.
42

Sayfa 44

Şekil 9. cEconomy bileşenleri arasındaki ilişki.
5.2. cEkonomi Tasarımı
Yukarıda belirtilen ödünleşimleri dengelemek için bir kriptoekonomi paketi öneriyoruz
cEconomy adı verilen ve birbirine sıkı sıkıya bağlı üç bileşeni içeren mekanizmalar:
Likidite Taahhüdü Kanıtı (PoLC) madenciliği, Likidite Destekleme Müzayedesi (LiBA)
ve Devlet Koruyucu Ağı (SGN). Üç bileşen arasındaki ilişki
Şekil 9'da gösterilmiştir.
Bu bileşenlerin ayrıntılarına geçmeden önce, ilk olarak birkaç
bu bölümde kullanılacak terimler. Özellikle, cEconomy'deki bir kullanıcı
sistem şu üç rolden herhangi birini oynayabilir: Zincir Dışı Hizmet Sağlayıcı (OSP), Son Kullanıcılar (EU), Network Liquidity Backer (NLB) ve State Guardians (SG). Zincir dışı Hizmet Sağlayıcılar (OSP), yüksek oranda yedekli çalıştırma teknik yeteneğine sahip kuruluşlardır, ölçeklenebilir ve güvenli zincir dışı altyapılar. Son Kullanıcılar (AB) zincir dışı erişebilir
OSP tarafından sağlanan hizmetler (örneğin, kripto para birimini öde ve al). Ortak olabilirler
tüketiciler veya IoT cihazları, VPN sağlayıcıları, canlı video akışı sağlayıcıları olabilirler
ve CDN sağlayıcıları, Makineden Makineye (M2M) sistemlerdeki karşı taraflar ve hatta
zincir dışı / zincir üstü akıllı sözleşme. Network Liquidity Backers (NLB) varlıklardır
zincir dışı infra operasyonlarını desteklemek için likiditesini sistemde kilitleyen
yapı. Eyalet Koruyucuları, AB'ye merkezi olmayan, güvenli, esnek
43

Sayfa 45

ve State Guardian Network aracılığıyla verimli devlet koruma hizmeti.
5.2.1. Likidite Taahhüdü Kanıtı (PoLC) Madenciliği
İlk hedefimiz, likiditeyi düşürerek ölçeklenebilirlik-likidite dengesini dengelemek.
Teknik olarak yetenekli tarafların zincir dışı hizmet sağlayıcıları olmalarının önündeki engel ve dolayısıyla
iyi ve güvenilir zincir dışı hizmetler için verimli ve rekabetçi bir pazar yaratmak.
Fikrin özü, hizmet sağlayıcıların büyük miktarlarda likiditeden yararlanmasını sağlamaktır.
ne zaman ihtiyaç duyarlarsa. Bu fikri gerçekleştirmenin ilk kısmı, bol miktarda
ve kısa vadeli likidite arz dalgalanmasını düzelten istikrarlı likidite havuzu.
Bu amaçla, Proof of Liquidity Commitment (PoLC) sanal madenciliği öneriyoruz
süreç.
PoLC madenciliği süreci, yüksek bir seviyeden, Ağ Likiditesini teşvik etmektir.
Destekçiler (NLB) likiditelerini kilitlemek için (dijital varlıklar şeklinde olabilir,
Kripto para birimleri ve CELR dahil ancak bunlarla sınırlı olmamak üzere Celer Network'te uzun süre
CELR tokenleri ile ödüllendirerek ve dolayısıyla istikrarlı ve
bol likidite havuzu.
Daha spesifik olarak, madencilik süreci, NLB'lerin boşta kalmalarını taahhüt etmelerini
(kilitlemelerini) içerir.
Teminat Taahhüt Sözleşmesi adı verilen "aptal kutuya" likidite (örneğin, ETH)
(CCC), belirli bir süre için. Bu süre zarfında dijital varlıkların
kilitlendiğinde, NLB'nin varlıkları yalnızca likidite destekleme sürecinde kullanılabilir ve
başka hiçbir şey. Daha resmi olarak, PoLC madencilik süreci şu şekilde tanımlanabilir.
Tanım 5.1 (PoLC Gücü). NLB i yerel kripto para biriminin S i miktarını kilitlerse

T bir blockchain (örneğin ETH) içerisinde i zaman, onun PoLC güç M i olarak hesaplanmıştır
M ben = S ben × T ben .
(18)
Tanım 5.2 (PoLC Teşvik Mekanizması). Sınırlı bir süre için, Celer
Ağ, kendilerini kilitleyen NLB'lere CELR şeklinde teşvikler sağlamayı amaçlamaktadır.
Sistem için bir destek gösterisi olarak CCC. Teşvikler orantılı dağıtılacaktır
her NLB'nin PoLC gücüne. R, i'nin teşviklerini göstersin, birinin sahip olduğu:
R ben =
R × M i
N
∑
j = 1
M j
,
(19)

Sayfa 46

burada R, mevcut blok için toplam ödüldür.
CCC'de likiditenin kilitlenmesinin herhangi bir doğal karşı taraf riski taşımadığını unutmayın.
basitçe Celer Network'e bir likidite taahhüdünü gösterir. Ayrıca, erken unutmayın
CCC'nin kilidinin açılmasına izin verilmez. Bir kişi bir "sahte tasfiye" oluşturmaya çalışabilir.
sahte bir OSP'nin “hacklenmesi” nedeniyle kişinin CCC'sinin sıvılaşması görüntüsü. Önlemek
Bu sahtekarlık, yeni çıkarılan CELR, şu tarihe kadar geri çekilemez ve kullanılamaz:
CCC kilidi açar. Herhangi bir erken tasfiye, halihazırda mayınlı CCC'nin kaybedilmesine neden olacaktır.
ve diğer madencilere yeniden dağıtıldı. Ortak bir likidite paydasının yapısı
PoLC'de de önemli bir sorudur. Platformun ilk lansmanı için
hedef blok zincirinin yerel para birimini kullanın ve daha sonra daha heterojen kullanın
dış fiyat oracle'ları aracılığıyla kripto varlıkları.
Bu mekanizmalar yürürlükte olduğunda, PoLC madencilik süreci, PoLC'nin
Sistemdeki güç, CELR'nin sistemi ve faydası büyüdükçe artacaktır.
pozitif bir döngü.
Bu noktada, CELR'nin böyle davranabilmesi için neden değerli olduğu merak edilebilir.
teşvik mi? Likidite Desteğini açıklayan aşağıdaki bölümlerde bunu açıklıyoruz
Müzayede ve Eyalet Koruyucu Ağları.

5.2.2. Likidite Destekleme Müzayedesi (LiBA)

Likidite bulmacasını çözmenin ikinci kısmı, zincir dışı hizmetler için bir yol sağlamaktır.
Likidite yoluyla elde edilen küresel likidite havuzuna erişim sağlayan sağlayıcı yardımcıları
Destek Müzayedesi (LiBA). LiBA, zincir dışı hizmet sağlayıcıların likidite talep etmesini sağlar
“kalabalık ödünç verme” yoluyla. Temelde, zincir dışı bir hizmet sağlayıcı bir LiBA başlatır.
Celer Network, belirli bir süre için belirli miktarda likidite “ödünç almak”.
İlgilenen bir likidite destekçisi, faiz oranını içeren bir teklif sunabilir.
teklif edilen likidite miktarı ve yatırmak istediği CELR miktarı
söz konusu süre. Likidite miktarı bir CCC aracılığıyla gönderilebilir. Yani,
CCC, likidite destekleyici bir varlık olarak hareket etme işlevine sahiptir. Ödünç alınan likidite
sahtekarlığa dayanıklı tahvil veya giden kanal depozitosu olarak kullanılacaktır.
LiBA, genelleştirilmiş çok özellikli bir Vickrey-Clarke-Groves'tur (kapalı teklif ikinci
puan) açık artırma. Açık artırma sürecini başlatmak için OSP, standart bir LiBA sözleşmesi oluşturur
Celer Network'ün merkezi LiBA kaydı aracılığıyla toplam
talep edilen likidite miktarı (q), talebin süresi (d) ve en yüksek faiz

---

Sayfa 47

kabul edebileceği oran (r max). Kaydı izleyen NLB'ler bu yeni LiBA'yı fark edecekler
sözleşme ve ihale sürecini başlatabilir. Celer Network tüm kripto varlıklarını gerektirir
ihale süreci için CCC'de kilitlenecek. CCC'nin "kilitsiz" olabileceğini ve
PoLC madenciliğinin işlevselliği olmadan basitçe bir destekleyici varlık olarak kullanılır. CCC şu
şekilde hareket eder:
kripto varlıkları için bir konteyner ve birleşik bir doğrulanabilir heterojen değer sağlar
kripto varlıkları. Ayrıca, CCC'nin kullanımı NLB'lerin katılmasını kolaylaştırır.
LiBA, kripto varlıklarını her teklif verdiklerinde hareket ettirmeden ve böylece
destek süreci ve güvenliği artırın. NLB i teklifi bir demet şeklinde sunar
b ben = (r ben , t ben , c i), burada r i faiz oranıdır, t i , istekli olduğu toplam CELR miktarıdır
sözleşme süresi boyunca kilitlenir ve c i , içinde bulunan toplam para birimi değeridir.
bu teklife bağlı CCC seti. Teklif verildikten sonra, ilgili
CCC'ler geçici olarak dondurulur. Mühürlü teklif verdikten sonra, LiBA sözleşmesi tersini kullanır
Aşağıdaki üç adımda kazanan teklifleri belirlemek için ikinci puan açık artırması [8].
• (Puanlama Kuralı). B = {b 1 , b 2 , ..., b n } teklif kümesindeki her teklif için b i = (r i , t i , c i)
f ben = t ben
c ben
, puanı s i şu şekilde hesaplanır:
s (b ben) = w 1
f ben
f max
- w 2
r ben
r max
,
(20)
burada f max = max {f 1 , f 2 , ..., f n } ve r max = max {r 1 , r 2 , ..., r n }. w , 1 ve w 2 olan
iki bileşenin ağırlıkları ve başlangıçta bir
daha yüksek ağırlıklı faiz oranı ve ardından stake edilen tutarı hesaba katın
CELR. 5 .
• (Kazanan Belirleme). Kimin olma fırsatına sahip olduğunu belirlemek için
ağ likidite destekçisi, LiBA sözleşmesi, teklifleri B'deki azalan sırada sıralar
puanlarına göre. Sıralanmış teklif seti B ∗ = {b ∗ ile gösterilir
1 , b ∗
2 , b ∗
3 , ..., b ∗
n }, nerede
s (b ∗
1) ≥ s (b ∗
2) ≥ ⋯ ≥ s (b ∗
n) (bağlar rastgele bozulur). Kazananlar ilk K teklifleridir
B * ,
K
∑
i = 1
t ben ≥ q ve
K − 1
∑
i = 1
t ben <q.
• (İkinci Skor CELR Staking / Tüketim). Kazananlar belirlendikten sonra,
onların CCC'leri LiBA sözleşmesinde d zamanı için kilitlenecektir (
talep), faiz talepleri kabul edilir ve faizler OSP tarafından önceden ödenir
likidite talebini başlatmak. Ancak, bunların hepsinin
taahhüt edilen CELR bu sözleşmede kilitlenir / tüketilir. Her kazananın yalnızca yapması gereken

5 Bu ağırlıklar gelecekteki merkezi olmayan yönetişim ayarlamasına tabidir.
46

Sayfa 48

kilitleyin / yeteri kadar CELR tüketin, böylece puanı ilk kaybedenin skoruyla eşleşir
bu açık artırma. Jetonun kilitlenip kilitlenmeyeceği veya tüketilip tüketilmeyeceği aşamaya bağlıdır
platformun. İlk beş yılda, PoLC aracılığıyla yeni tokenler üretilecek
madencilik ve LiBA yalnızca belirteç stake etmeyi gerektirir. PoLC madenciliği sona erdiğinde,
LiBA, jetonu tüketmeye başlayacak ve tüketilen jetonlar,
sistem sürekli PoLC madencilik ödülleri olarak. İkinci puanın altında
CELR stake / tüketim mekanizması, katılımcıların sunması öngörülüyor
malın gerçek değerine (doğruluk [17]) uyan teklifler (bu durumda,
ağ likiditesini destekleme fırsatı).
Örnek: Bir OSP'nin aşağıdaki parametrelerle bir LiBA başlattığını varsayalım
(600 ETH, 30 gün,% 1) ve üç potansiyel teklif veren var (diyelim ki A, B ve
C) bu LiBA için. Üç teklif sahibinin teklifi şu şekildedir: b A = (% 1, 800 CELR, 400 ETH); b B
= (% 0,5, 800 CELR, 200 ETH); b C = (% 1, 100 CELR, 400 ETH). Göre
Puanlama kuralı, s B > s A > s C var . A ve B tüm talebi karşılayabileceğinden,
kazanan olarak seçilirler. A ve C'nin sahip olmasına rağmen
aynı faiz oranı (% 1) ve aynı miktarda likidite (400 ETH) sağlamak,
Teklif veren C kaybederken, teklif veren A kazanan olarak seçilir; bu gerçeğinden kaynaklanmaktadır
bu platforma katkılarının bir sembolü olarak taahhüt edilmiş CELR tokenleri,
önemli ölçüde farklıdır. Son olarak, ikinci skor stake etme kuralına göre, A
ve B CELR jetonlarını 30 gün boyunca C puanına uyacak şekilde kilitler (veya tüketir).
Müzayede süreci bittikten sonra likidite talebini başlatan OSP öder
LiBA sözleşmesine para yatırarak kazanan likidite destekçilerinin faizleri.
Faiz ödemesini aldıktan sonra, LiBA sözleşmesi faizleri verir
karşılık gelen likidite destekçileri ve 1: 1 destekli cETH'ler çıkarır (ETH'yi
bir örnek) likidite talep tutarıyla eşleşen. CETH esasen
bir IOU, bu IOU'lar ağ tarafından% 100 sigortalandığından kullanıcı için hiçbir risk oluşturmaz
LiBA sözleşmesinde likidite destekçileri.
Normal durumlarda, LiBA sözleşmesi, OSP'nin zaman aşımından önce çözülür.
tüm cETH jetonlarını geri gönderir. Temel olarak, zaman aşımından önce OSP tüm
yukarı akış kanallarından toplanarak gerçek ETH'lerle AB'lere ödenen cETH'ler
tively.
47

Sayfa 49

OSP'nin saldırıya uğraması durumunda, Celer Network'ün güven modeli değişebilir.
Herhangi bir protokol düzeyinde ek yükü olmayan en basit güven modeli itibara dayalıdır,
NLB'lerin herhangi bir varsayılan geçmişi olmayan saygın bir OSP seçtiği yerlerde. Bu kadar basit
NLB'ler, CCC'leri olduğu için fon ve varlıkları kaybetme riskine maruz kalmaktadır.
OSP'nin temerrüde düşmesi halinde AB'ler için sigortalar. Bununla birlikte, tartışılabilir bir durumdur
ki bunda bile
basit güven modeli, son derece güvenilir ve itibarlı bir OSP çalıştırmak mümkündür; bu
tüm desteklerin kaybolması pek olası değildir. Ek güvenlik özellikleri var
Potansiyel riski daha da hafifletmek için LiBA'nın etrafına eklenebilir. Örneğin,
yeni çıkarılan cETH'lerin yalnızca eyalet kanalının bir beyaz listesine yatırılmasına izin verilir
sözleşmeler; cETH'lerin yalnızca artımlı olarak bir üst sınırla kullanılmasına izin verilir
harcama hızı. Ayrıca bir OSP'nin güvenliğini sağlamak için yapabileceği birçok şey vardır.
bölümlere ayrılmış çok düğümlü dağıtım, resmi doğrulama gibi altyapı
ağ altyapısının güvenlik erişim kuralı ve daha fazlası.
Ek olarak, rastgele seçilmiş bir quo-
NLB'nin bir OSP'nin operasyonlarını (örneğin ödeme) birlikte imzalaması gerekecektir. Bu NLB'ler

yalnızca ve yalnızca ile gelen bir işlem görürlerse giden bir aktarıma izin ver
eşleşen miktar. Bu NLB'ler ayrıca OSP'nin gelen ödemelerine de bağlıdır. Eğer
OSP sonunda geri ödemeyi yapamazsa, NLB'ler için birinci öncelik hakkı olacaktır.
diğer kanallardan OSP'ye gelen fonları talep edin. Ancak, bunun
işletim modeli, kaçınılmaz olarak, ağın bir miktar verimliliğinden ödün verecektir.
Bunları söyledikten sonra, güven modelindeki nihai dengenin olması gerektiğine inanıyoruz.
piyasa talebi ile tanımlanır. Pazar için her iki güven modelini de organik olarak açıyoruz.
gelişmek. İlk günlerde güvensiz modelin daha uygun olacağını düşünüyoruz.
ağın başlatılması ve daha sonra daha güvene dayalı hale gelecektir.
LiBA'nın güven modelinden bağımsız olarak, LiBA sürecinin
Son kullanıcıların gerekli likidite olarak hiçbir zaman güvenlik riski almamasını sağlar
LiBA sözleşmesi ile% 100 "sigortalıdır". Celer Network'ün sisteminde,
Yardımsever son kullanıcıların güvenlik konusunda endişelenmelerine gerek olmadığından emin olun.
aldıkları fon ve LiBA bunu başardı. PoLC ve LiBA birlikte bir
bol likidite havuzu, zincir dışı hizmet sağlayıcı olmanın önündeki engelleri düşürür,
merkezileştirme riskini azaltın ve ağın benimsenmesini hızlandırın.

Sayfa 50

5.2.3. State Guardian Ağı
CELR'nin diğer bir kullanımı, yeni sigorta ile zincir dışı veri kullanılabilirliği sağlamaktır.
ölçeklenebilirlik-kullanılabilirlik ödünleşimlerini dengeleyen model ve basit etkileşimler
Bölüm 5.1.2'de belirtildiği gibi.
Yüzeyden bakıldığında, kullanılabilirlik sorunu çözülmesi kolay bir sorun gibi görünüyor. Bir
bu sorunun olası cevabı şu olabilir: hadi bazı izleme hizmetleri oluşturalım
gelecek ve insanlar kendileri olmadıklarında bu izleme hizmetleri için ödeme yapacaklardır.
internet üzerinden. İlk bakışta mantıklı bir çözüm gibi geliyor ama biz bu düşünce zincirini sürüyoruz
biraz ileride, izleri bozan kusurları hemen göreceğiz.
Şu soruyla başlayalım: bu izleme hizmetleri güvene dayalı mı? Eğer
cevap evet, o zaman başka bir merkezi tıkanma noktası, tek bir başarısızlık noktası yaratır
ve güvenli değil. Kötü niyetli karşı taraf bu izleme hizmetlerine kolayca rüşvet verebilir
hayırsever son kullanıcılara zarar vermek.
Güvensiz bir izleme hizmeti oluşturabilir miyiz? Örneğin, biz
Kullanıcılar için eyaletleri savunamazlarsa izleme hizmeti sağlayıcılarını cezalandırırlar.
Bununla birlikte, bu fikri incelerken, hemen işleyen bazı uyarıları görüyoruz.
bu yaklaşım pratik değildir. İzleme hizmeti sağlayıcıları ne kadar ceza ödemelidir?
Sürtüşmeleri göz ardı ederek, hizmet sağlayıcıları izlemek için toplam ceza teminatı,
Çevrimdışı olan tarafın maruz kaldığı en büyük potansiyel kayba eşit olmalıdır.
Bu, zincir dışı bir ağ için likidite gereksinimini etkili bir şekilde ikiye katlar çünkü
Kanallardaki mevcut kilitli likiditeye ek olarak, birisi çevrimdışı olduğunda
veya yan zincirlerde sahtekarlığa dayanıklı bağ, izleme hizmeti sağlayıcılarının da kilitlenmesi gerekir
ceza mevduatı ile aynı miktarda likidite.
Daha da kötüsü, izleme hizmeti sağlayıcılarının farklı varlıklar için farklı varlıkları tutması gerekir.
ilgili izleme görevleri ve işler gerçekten karmaşık hale gelebilir.
karmaşıktır ve birden fazla varlık sınıfı işin içindedir. Bazen bir
tüm karmaşıklık göz önüne alındığında, durumdan temel değere doğrudan çeviri
genelleştirilmiş durum kanalları için durum bağımlılığı.
Yeterli likidite olsa bile, buradaki "sigorta" modeli gerçekten katıdır:
temelde izleme hizmeti sağlayıcıları başarısız olursa% X oranında geri aldığınızı söylemek
eyaletlerinizi savunmak için. Büyük bir X değeri seçerseniz, gerçekten pahalı hale gelebilir
ek likidite kilidi nedeniyle, ancak küçük bir X değeri seçerseniz,
gerçekten güvensiz hale geliyor.

Sayfa 51

Bu dezavantajların yanı sıra, devlet izleme hizmetlerinin fiyatının nasıl olduğu belirsizdir.
Pazar bilgileri hala düşük verimlilikle ayrıldığı için belirlenmelidir.
Bu düşük verimlilik ve heterojen varlıklar üzerindeki parti başına tahvil,
izleme hizmetleri ve parçalama ile karmaşık zincir içi ve zincir dışı etkileşimler
herhangi bir zincir dışı platformun kullanılabilirliği. Daha fazla sorun var, ancak yukarıdakiler zaten yeterince kötü.
Bu sorunları çözmek için State Guardian Network'ü (SGN) öneriyoruz. Devlet Muhafızı
Ağ, kullanıcılar çevrimdışıyken zincir dışı durumları korumak için özel bir kompakt yan zincirdir.
CELR token sahipleri, CELR'lerini SGN'ye yatırabilir ve eyalet koruyucusu olabilir.
Bir kullanıcı çevrimdışı olmadan önce durumunu SGN'ye belirli bir ücret karşılığında gönderebilir ve veliler, devletini belirli bir süre korumak için. Bir dizi koruyucu
daha sonra bu durumdan sorumlu olacak şekilde rastgele seçilir ve durum karması ve
"Sorumluluk puanı". Velilerin seçilmesine ilişkin ayrıntılı kurallar aşağıdaki gibidir.
• (Devlet koruma talebi). Bir durum koruma talebi bir demettir $\eta_i = (s_i, l_i, d_i)$
Nerede $s_ı$ l korunmalıdır durumdur i hizmet ücreti miktarı ödenir
velilere ve d_i, bu devletin korunması gereken süredir.
• (Sorumluluk Puanı). Devlet koruma talebinin sorumluluk puanı η_i
şu şekilde hesaplanır:

$$\gamma_i = \frac{l_ı}{d_{ben}}.$$

Bir kullanıcının Sorumluluk Puanı esasen bunun yarattığı gelir akışıdır.
kullanıcı SGN'ye.
• (Veli bahislerinin sayısı). Bir dizi olağanüstü devlet koruması verildiğinde
istek $R = \{\eta_1, \cdots, \eta_m\}$, her istek için söz konusu CELR sayısı $\eta_i \in R$
dır-dir

$$n_i = \gamma_{ben} \frac{m}{\sum_{j=1} \gamma_j} K,$$

burada K Koruyucuların SGN'de hissettiği CELR hisselerinin toplam sayısı. İçinde
başka bir deyişle, hisseye konu olan sorumlu CELR miktarı oranla orantılıdır.
bu talepler arasında sorumluluk puanı tüm bekleyen durumların toplamına eşittir

---

Sayfa 52

sorumluluk puanları.
• (Vasi hisselerinin atanması). H devlet koruma isteği göz önüne alındığında i , h let i olmak
karşılık gelen durum s_i için karma değeri (örneğin, Keccak256 karması). Her bir CELR
pay k, bir KIMLIK p_k ile ilişkilidir (bu aynı zamanda bir karma değerdir). $\Delta(g_1, g_2)$ olsun
iki hash değeri arasındaki mesafe g_1 ve g_2 (örneğin, kullanılan mesafe ölçüsü
Akor DHT [14]). Daha sonra CELR bahisleri, değerlerine göre artan sırada sıralanır.
hash değerine olan uzaklık h_i. Varsayalım ki $\delta(p_1, h_i) \leq \delta(p_2, h_i) \leq \cdots \leq \delta(p_K, h_i)$
(bağlar rastgele bozulur). En küçük para birimine sahip ilk n_i CELR hissesi
mesafe seçilir ve ilgili hissenin sahibi eyalet olur
bu istek için vasi.
(Devlet Koruma Hizmeti Ücret Dağıtımı). Her devlet koruma talebi için $\eta_i =$
(s_i, l_i, d_i), ekli hizmet bedeli l_i,
aşağıdaki kural. Her eyalet koruyucusu j için, z_j hisseli CELR'si olsun.

bu eyalet koruma talebi için seçilmiş olanlar. Sonra velinin hizmet bedeli
η j devlet onu koruyor isteğinden alır i ise
q j =
z j × l ben
n ben
.
Her hisseli CELR'nin bir durum için aynı seçilme olasılığına sahip olduğuna dikkat edin
koruma talebi. Sonuç olarak, bir SG'nin bakış açısına göre, CELR, SGN'de o kadar çok hisseye sahip olur,
Bu tür SG en kazıklar daha beklenti seçilebilir (yani, Z değeri j olacak
daha büyük), dolayısıyla alacağı hizmet bedelleri artacaktır. Sağlayan
CELR, SGN üyeliği olarak önemli bir değerdir.
(Güvenlik ve Gizli Anlaşmaya Karşı Direniş). Her veliye bir anlaşmazlık yuvası atanır
yerleşim zaman aşımına göre. Vasi, gerektiğinde yuvasına itiraz edemezse
için, sonraki veliler olayı bildirebilir ve başarısız vasinin CELR'sini alabilir
kazık. Sonuç olarak, seçilen velilerden en az biri bozulmadığı sürece
ve işi yerine getirdiğinde, bir son kullanıcının durumu her zaman güvenlidir ve anlaşmazlığa açıktır.
SGN mekanizması ayrıca aşağıdaki ek değerleri de beraberinde getirir.
• Vasiler için önemli bir likidite kilidi gerektirmez. Muhafızlar
sadece keyfi durumları korumak için kullanılabilecek CELR'lerini destekliyorlar.
temel değerin / belirteçlerin türü / miktarından daha az.

Sayfa 53

• Keyfi durum izleme için birleşik bir arayüz sağlar. Ne olursa olsun
devletin ETH, herhangi bir ERC20 belirteci veya karmaşık durumlar ile ilgili olup olmadığı,
kullanıcılar sadece bir ücret ekler ve bunu SGN'ye gönderir. SGN umursamıyor
temel devletler ve ilgili değer ve basitçe miktarını tahsis eder
Devletten sorumlu olmak için ödenen ücretle orantılı CELR.
• Basit etkileşimler sağlar. Celer Network kullanıcılarının iletişim kurmasına gerek yoktur.
bireysel veliler ve sadece bu yan zincire durumları sunmaları gerekir.
• En önemlisi, tamamen yeni ve esnek bir devlet korumasını mümkün kılar-
ekonomik dinamikler. Katı ve opak olanı zorlamak yerine "% X kazanın
eski model SGN, kullanıcılara "paramı geri almak için yeni bir mekanizma getiriyor.
X dönemi "ve bu değişken sigorta için verimli bir fiyatlandırma mekanizması
model. Risk altındaki tüm veliler bir kullanıcı için itiraz edemezse, CELR alacak
tazminat olarak bu velilerden alınan hisseler. Kararlı durumda, CELR belirteçleri
SGN'de stake edilmiş olanlar, gelen bir akışı temsil eder (örneğin, x Dai / saniye kazanma).
Bir kullanıcı gönderirken durum izleme maliyetini ve diğer sürtüşmeleri göz ardı etmek
devletten SGN'ye, CELR'nin ne kadarını "kapsadığını" açıkça seçebilir
saniye başına ödenen ücretleri (yani sorumluluk puanı) seçerek durumu.

5.2.4. Özet

Sistematik olarak düşünen cEconomy, zincir dışı bir platformun tüm yaşam döngüsünü kapsar.
LiBA ve PoLC madenciliği, ara işlemleri zincir dışına çıkarmakla ilgilidir.
düşük engelli moda. SGN, en güncel bilgileri sunma yeteneğini güvence altına almakla ilgilidir.
Gerektiğinde tekrar zincirde olduğunu belirtir. Bu nedenle, cEconomy'nin ilk şirket olduğuna
inanıyoruz.
Yeni değer getiren ve diğerlerini mümkün kılan yoğun zincir dışı platform kripto ekonomisi
bilge imkansız dinamikler.

6. Sonuç

Celer Network, İnternet'i getiren tutarlı bir teknoloji ve ekonomik mimaridir.
mevcut ve gelecekteki blok zincirleri için ölçeklenebilirlik düzeyi. Yatay olarak ölçeklenebilir, güven-
ücretsiz, merkezi olmayan ve gizliliği koruyan. Katmanlı bir mimariyi kapsar.
her katmanda önemli teknik yenilikler. Ek olarak, Celer Network,
elde etmek için yapılan ödünleşimleri dengelemek için ilkeli bir zincir dışı kriptoekonomi tasarımı

Sayfa 54

ölçeklenebilirlik. Celer Network, blockchain ve
merkezi olmayan uygulamaların oluşturulma ve kullanılma biçiminde devrim yaratın.
7. Lider Geliştiriciler
Celer Network, iyi eğitilmiş dağıtılmış bir sistem ve ağdan oluşan bir ekip tarafından geliştirilmiştir.
bazılarında çalışan sistem araştırmacıları ve deneyimli mühendisler çalıştı.
en zorlu dağıtık sistem ve ağ mimarisi tasarım sorunları ve
ürün.
Dr. Mo Dong, doktora derecesini aldı. UIUC'tan. Araştırması, öğrenmeye dayalı
ağ protokolü tasarımı, dağıtılmış sistemler, resmi doğrulama ve Oyun Teorisi.
Dr. Dong, İnternet TCP'sinde devrim yaratan ve kıtalar arası iyileştirilen projeye öncülük etti
pişmanlık duymayan öğrenme algoritmaları ile 10X ila 100X arasında veri aktarım hızı. Onun işi
en iyi konferanslarda yayınlanan, Internet2 Yenilikçi Uygulama Ödülü'nü kazandı ve
başlıca İnternet içeriği ve servis sağlayıcıları tarafından benimsenmiştir. Dr. Dong bir kurucuydu
Veriflow'da mühendis ve ürün yöneticisi, ağ resmi konusunda uzmanlaşmış bir başlangıç
doğrulama. Geliştirdiği resmi doğrulama algoritmaları ağları koruyor
servet 50 şirket için güvenlik. Dr. Dong ayrıca Algoritmik uygulamada deneyimlidir.
Oyun Teorisi, özellikle açık artırma teorisi, bilgisayar sistemi protokol tasarımlarına. O sahip
tam kapsamlı akıllı sözleşme kursları öğretiyor. Teknik bloglar ve videolar üretiyor
7000'den fazla aboneye sahip blok zincirinde.
Dr. Junda Liu, doktorasını aldı. UC Berkeley'den Prof. Scott Shenker danışmanlığında.
Nanosaniye elde etmek için DAG tabanlı yönlendirme öneren ve geliştiren ilk kişiydi.
ağ kurtarma (son teknolojiye göre 1000 kat iyileştirme). Dr. Liu, 2011'de Google'a katıldı
öncü araştırmasını Google'ın küresel altyapısına uygulamak. Teknoloji lideri olarak, o ...
1000 terabit / s bant genişliği ikiye bölme kapasitesine sahip dinamik bir veri merkezi topolojisi
geliştirdi
ve 1 milyondan fazla düğümü birbirine bağlayarak. 2014 yılında Dr. Liu bir kurucu oldu
Project Fi üyesi (Google'ın yenilikçi mobil hizmeti). O teknoloji lideriydi
sorunsuz taşıyıcı geçişi ve bir konseptten 100 milyon $ / yıl üzerinde bir işletmeye geçişi denetleyen Fi
2 yıl içinde. Ayrıca, operatör hizmetleri için Android Teknoloji Lideri idi.
1,5 milyardan fazla cihaz. Dr. Liu 6 ABD patentine sahiptir ve çok sayıda makale yayınlamıştır.
en iyi konferanslarda. Tsinghua Üniversitesi'nden BS ve MS aldı.
53

Sayfa 55

Dr. Xiaozhou Li, doktorasını aldı. Princeton Üniversitesi'nden ve genel olarak birbirleriyle
dağıtılmış sistemler, ağ oluşturma, depolama ve veri yönetimi araştırmalarında değerlendirilmiştir. O
SOSP, NSDI, FAST, SIGMOD, EuroSys, CoNEXT dahil olmak üzere en iyi mekanlarda yayınlıyor,
ve dağıtılmış bir koordinasyon hizmeti oluşturarak NSDI'18 en iyi kağıt ödülünü kazandı
milyarlarca QPS işlem hacmi ve on mikrosaniye gecikme süresiyle. Xiaozhou uzmanlaşmıştır
düşük seviyede yüksek performans sağlayan ölçeklenebilir algoritmalar ve protokoller geliştirmede
maliyet, bazıları yaygın olarak kullanılan sistemlerin temel bileşenleri haline gelmiştir.
Google TensorFlow makine öğrenimi platformu ve Intel DPDK paket işleme olarak
çerçeve. Xiaozhou, bir başlangıç şirketi olan Barefoot Networks'te çalıştı.
dünyanın en hızlı ve en programlanabilir ağları, birçok çığır açan
projeler, kilit müşterilerle teknik ilişki kurdu ve altı ABD patent başvurusu yaptı.
Qingkai Liang, doktorasını aldı. dağıtılmış alanında MIT'den derece
rakip ortamlarda optimum ağ kontrol algoritmalarında uzmanlaşmış sistemler
ments. İlk olarak 15'in üzerinde üst düzey makale yazdı ve 5 yüksek performanslı ve
başarılı bir şekilde uygulanan son derece sağlam, düşmanca dirençli yönlendirme algoritmaları
Raytheon BBN Technologies ve Bell Labs gibi sektörde çalıştı. O oldu
IEEE MASCOTS 2017'de En İyi Makale Adayı ve Oturumun En İyisi

IEEE INFOCOM 2016 ve 2018'de Sunum Ödülü.
54

Sayfa 56

Referanslar
[1] Plazma: https://plasma.io/plasma.pdf,.
[2] Raiden Ağı Belgeleri: https://raiden-network.readthedocs.io. Ocak-
uary 2018.
[3] R. Khalil ve A. Gervais, Revive: Off-Blockchain Ödeme Ağlarının Yeniden Dengelenmesi, Pro-
2017 ACM SIGSAC Bilgisayar ve İletişim Güvenliği Konferansı'nın temelleri
rity. ACM, 2017, s. 439–453.
[4] G. Malavolta, P. Moreno-Sanchez, A. Kate ve M. Maffei, SilentWhispers: Enforcing
kredi ağlarında güvenlik ve mahremiyet, NDSS, 2017.
[5] A. Miller, I. Bentov, R. Kumaresan ve P. McCorry, Sprites: Payment chan-
Yıldırımdan daha hızlı giden nels, CoRR abs / 1702.05812 (2017). Mevcut
http://arxiv.org/abs/1702.05812.
[6] P. Moreno-Sanchez, A. Kate, M. Maffei ve K. Pecina,
kredi ağları, Ağ ve Dağıtık Güvenlik Sempozyumu. 2015.
[7] MJ Neely, E. Modiano ve CE Rohrs, Dinamik güç tahsisi ve zaman için yönlendirme-
çeşitli kablosuz ağlar, IEEE Journal on Selected Areas in Communications 23 (2005),
s. 89–103.
[8] L. Pham, J. Teich, H. Wallenius ve J. Wallenius, Çok nitelikli çevrimiçi ters müzayedeler:
Son araştırma eğilimleri, European Journal of Operational Research 242 (2015), s. 1-9.
[9] J. Poon ve T. Dryja, Bitcoin lightning ağı: Ölçeklenebilir zincir dışı anlık ödemeler,
Teknik Rapor (taslak) (2015).
[10] P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy ve O. Osuntokun, Flare: Bir yaklaşım
yıldırım ağında yönlendirmeye (2016).
[11] MG Reed, PF Syverson ve DM Goldschlag, Anonim bağlantılar ve soğan
yönlendirme, IEEE Journal on Selected Areas in Communications 16 (1998), s. 482–494.
[12] S. Roos, M. Beck ve T. Strufe, Verimli ve esnek yönlendirme için Anonim adresler
F2F kaplamalarında, Bilgisayar İletişiminde, IEEE INFOCOM 2016 - 35. Yıllık
IEEE Uluslararası Konferansı. IEEE, 2016, s. 1–9.
[13] S. Roos, P. Moreno-Sanchez, A. Kate ve I. Goldberg, Ödemeleri hızlı bir şekilde ödeme ve
özel: Yol tabanlı işlemler için verimli merkezi olmayan yönlendirme, arXiv ön baskı
arXiv: 1709.05748 (2017).
[14] I. Stoica, R. Morris, D. Liben-Nowell, DR Karger, MF Kaashoek, F. Dabek ve H.
Balakrishnan, Chord: internet uygulamaları için ölçeklenebilir bir eşler arası arama protokolü,
Ağ Oluşturmada IEEE / ACM İşlemleri (TON) 11 (2003), s. 17–32.
[15] L. Tassiulas ve A. Ephremides, Kısıtlı kuyruk sistemlerinin kararlılık özellikleri ve
55

Sayfa 57

çok noktalı radyo ağlarında maksimum verim için planlama politikaları, IEEE işlem
otomatik kontrol 37 (1992), s. 1936–1948.
[16] PF Tsuchiya, The Landmark Hierarchy: Çok büyük ağda yönlendirme için yeni bir hiyerarşi
ACM SIGCOMM Computer Communication Review, Cilt. 18. ACM, 1988, s.
35–42.
[17] HR Varian ve C. Harris, Teorik ve pratikte vcg müzayedesi, American Economic
İnceleme 104 (2014), s. 442–45.
56