INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCHERS
ISSN: 3030-332X Impact factor: 9,493 Volume 17, issue 1, Aprel 2026
https://wordlyknowledge.uz/index.php/IJSR worldly knowledge
*Index: google scholar, research gate, research bib, zenodo, open aire.*
https://scholar.google.com/scholar?hl=ru&as_sdt=0%2C5&q=wosjournals.com&btnG
https://www.researchgate.net/profile/Worldly-Knowledge
https://journalseeker.researchbib.com/view/issn/3030-332X

# ANDROID MOBIL ILOVALARIDA TARMOQ O‘ZGARUVCHANLIGIGA MOSLASHADIGAN BATCH BOSHQARUVI VA CIRCUIT BREAKER USULLARI

**Ismoiljonova Maftuna Ibrohim qizi** ,talaba

Andijon davlat texnika instituti , Andijon, o‘zbekiston

Elektron pochta: ismoiloveimona@gmail.com

Telefon: +998951662101

**Dexqonov Tavakkaljon Alimjonovich** , Katta o‘qituvchi

Andijon davlat texnika instituti , Andijon, o‘zbekiston

Elektron pochta: dehqonovtavakkaljon@gmail.com

Telefon: +998 97 610 78 75

## ANNOTATSIYA

Sanoat va logistika sohasidagi Android mobil ilovaları tez-tez o‘zgarib turuvchi tarmoq muhitlarida ishlaydi: Wi-Fi dan 2G gacha o‘tishlar kutilmagan vaqt tugashi va ma’lumot yo‘qolishiga olib keladi. Statik batch boshqaruvi bu sharoitda samarasiz bo‘lib, tizim unumdorligini pasaytiradi. Ushbu maqolada HotOrNotOmbor SQLite tizimi bilan integratsiyalashgan Adaptiv Batch Boshqaruvi va Circuit Breaker naqshlarining kombinatsiyalangan yondashuvi taqdim etiladi. Adaptiv algoritm tarmoqning joriy holatiga qarab batch o‘lchamini 5 dan 100 gacha dinamik ravishda o‘zgartiradi. EWMA asosidagi latensiya teskari aloqasi tizimni to‘satdan tarmoq o‘zgarishlaridan himoya qiladi. Circuit Breaker naqshi uch holat automati orqali kaskad muvaffaqiyatsizliklarni oldini oladi.

**Kalit so‘zlar:** Adaptiv Batch, Circuit Breaker, Eksponent Orqaga Qaytish, Android WorkManager, SQLite, OkHttp, Kotlin Coroutines.

## ABSTRACT

Android mobile applications in the industrial and logistics sectors operate in frequently changing network environments: transitions from Wi-Fi to 2G lead to unexpected timeouts and data loss. Static batch management is ineffective in these conditions, reducing system performance. This paper presents a combined approach of Adaptive Batch Management and Circuit Breaker patterns integrated with the HotOrNotOmbor SQLite framework. The adaptive algorithm dynamically changes the batch size from 5 to 100 depending on the current network state. Latency feedback based on EWMA protects the system from sudden network changes. The Circuit Breaker pattern prevents cascading failures through three state machines.

**Keywords:** Adaptive Batch, Circuit Breaker, Exponential Backtracking, Android WorkManager, SQLite, OkHttp, Kotlin Coroutines.

## АННОТАЦИЯ

Мобильные приложения Android, используемые в промышленном и логистическом секторах, работают в условиях часто меняющейся сетевой среды: переходы с Wi-Fi на 2G приводят к неожиданным таймаутам и потере данных.

Статическое пакетное управление в таких условиях неэффективно, что снижает производительность системы. В данной статье представлен комбинированный подход, сочетающий адаптивное пакетное управление и паттерн «Автоматический выключатель», интегрированный с фреймворком HotOrNotOmbor SQLite. Адаптивный алгоритм динамически изменяет размер пакета от 5 до 100 в зависимости от текущего состояния сети. Обратная связь по задержке на основе EWMA защищает систему от внезапных изменений сети. Паттерн «Автоматический выключатель» предотвращает каскадные сбои с помощью трех конечных автоматов.

**Ключевые слова:** Адаптивное пакетное управление, Автоматический выключатель, Экспоненциальный возврат, Android WorkManager, SQLite, OkHttp, Kotlin Coroutines.

## I. KIRISH

Zamonaviy sanoat va logistika sohasida Android qurilmalari asosidagi mobil ilovalar ishlab chiqarish sexlaridan, ombor maydonlaridan va ochiq hududlardan foydalaniladi. Ushbu muhitlarda tarmoq ulanishi barqaror bo‘lmaydi: bir lahzada Wi-Fi orqali yuqori tezlikda ishlayotgan qurilma bir necha daqiqadan so‘ng 3G yoki 2G/Edge tarmoqqa o‘tib qolishi mumkin. Ba’zan esa umuman tarmoq bo‘lmaydi [1, 2].

Ko‘pgina mavjud ilovalar statik batch boshqaruvidan foydalanadi — ya’ni serverga yuboriladigan ma’lumotlar to‘plami o‘lchami doimiy qilib belgilanadi. Bu yondashuv yuqori tarmoq sifatida samarali bo‘lsa-da, past tarmoq sharoitida jiddiy muammolarni keltirib chiqaradi: timeout xatolari, ma’lumot yo‘qolishi va foydalanuvchi tajribasining yomonlashuvi [3, 4].

Parallel ravishda, server tomonidagi muammolar — yuklanish ortishi, tarmoq uzilishlari — kaskad muvaffaqiyatsizliklarni yuzaga keltirishi mumkin. Circuit Breaker naqshi bu muammoni hal qiluvchi dasturlash arxitekturasi sifatida keng e'tirof etilgan [5]. Biroq Adaptiv Batch va Circuit Breaker naqshlarining birgalikda, domen-spetsifik SQLite tizimi bilan integratsiyasi yetarlicha o‘rganilmagan.

Ushbu maqola quyidagi asosiy hissalarni qo‘shadi:

(1) tarmoq o‘tkazish qobiliyatiga asoslangan dinamik batch o‘lchami algoritmi;

(2) EWMA latensiya teskari aloqasi orqali spike-bardoshli tartibga solish mexanizmi;

(3) uch holatli Circuit Breaker avtomati va uning HotOrNotOmbor SQLite tizimiga integratsiyasi;

(4) haqiqiy sanoat muhitida o‘tkazilgan empirik baholash.

## II. ADABIYOTLAR SHARHI

Mobil tarmoq moslashuvchanligini o‘rganishda Flinn va Satyanarayanan [1] mobil qurilmalarda resurs moslashuvi kontseptsiyasini kiritgan. Ularning ishi o‘zgaruvchan tarmoq sharoitida ilovalar xatti-harakatini dinamik ravishda moslashtirish zarurligini ko‘rsatgan.

Batch ma’lumot uzatish sohasida Huang va boshqalar [3] Android ilovalarida tarmoq so‘rovlarini to‘plash (batching) strategiyalarini o‘rganib, energiya sarfini 30% gacha

kamaytirish mumkinligini namoyish etgan. Biroq ularning ishi tarmoq o‘tkazish qobiliyatining dinamik o‘zgarishini inobatga olmagan.

Circuit Breaker naqshi dastlab Nygard [5] tomonidan tavsiflanib, tarqatilgan tizimlarda kaskad muvaffaqiyatsizliklarini oldini olish uchun tavsiya etilgan. Netflix-ning Hystrix kutubxonasi [6] ushbu naqshni sanoat miqyosida joriy etishning eng mashhur namunasi hisoblanadi. Android muhitida esa Kotlin Coroutines va OkHttp interceptorlar orqali amalga oshirish usullari Gokhroo va boshqalar [7] tomonidan o‘rganilgan.

EWMA filtrlash texnikasi tarmoq ko‘rsatkichlarini silliqlashtirish uchun keng qo‘llaniladi [8]. Eksponent og'irlikli o‘rtacha yangi ma’lumotlarga ko‘proq og'irlik beradi, shu bilan birga tarixiy qiymatlarni inobatga oladi. Bu xususiyat uni to‘satdan spike lardan himoya uchun ideal qiladi. Offline ma’lumotlarni sinxronlashtirish va WorkManager integratsiyasi Jones va Becker [9] tomonidan Android arxitekturasi kontekstida ko‘rib chiqilgan. SQLite tranzaksiyalari va batch operatsiyalari Shasha va Bonnet [10] tadqiqotida batafsil tahlil qilingan.

## III. METODOLOGIYA

### A. Tizim arxitekturasi

Taklif etilgan yondashuv uchta asosiy komponentdan iborat: (1) AdaptiveBatch moduli — tarmoq holatini kuzatib, optimal batch o‘lchamini hisoblaydi; (2) CircuitBreaker moduli — server sog'lig'ini nazorat qilib, kaskad muvaffaqiyatsizliklarni oldini oladi; (3) HotOrNotOmbor SQLite konnektori — lokal ma’lumotlarni saqlash va batch ajratish uchun xizmat qiladi. Barcha komponentlar Android WorkManager orqali fon ishi sifatida boshqariladi va Kotlin Coroutines bilan asinxron ravishda ishlaydi.

### B. Adaptiv batch algoritmi

Adaptiv batch algoritmi ConnectivityManager API orqali tarmoq turini aniqlaydi va I-jadvalda ko‘rsatilgan qoidalar asosida optimal batch o‘lchamini belgilaydi.

**I-JADVAL — Tarmoq turiga qarab optimal batch o‘lchami**

| Tarmoq turi | Bandwidth | Batch o‘lchami | Asoslash |
|---|---|---|---|
| Wi-Fi / Ethernet | >= 10 Mbps | 100 (MAX) | To‘siqsiz, tez yuborish imkoni |
| 4G / LTE | >= 10 000 kbps | 50 | Yetarli tezlik, katta hajmlar mumkin |
| 3G | >= 1 000 kbps | 20 | o‘rtacha sifat, ehtiyotkorlik talab etiladi |
| 2G / Edge | < 1 000 kbps | 10 | Sekin ulanish, minimal batch |
| Tarmoq yo‘q | — | 5 (MIN) | Offline rejim, lokal bufer |

Tarmoq turi belgilangandan so‘ng, latensiya teskari aloqasi EWMA formulasi yordamida amalga oshiriladi:

L_yangi = 0,7 × L_eski + 0,3 × L_joriy

Bu yerda:

L_yangi — yangilangan latensiya bahosi (ms)

L_eski — avvalgi EWMA qiymati (ms)

L_joriy — oxirgi so‘rovning haqiqiy latensiyasi (ms)

0,7 — tarixiy og'irlik (beta = 1 - alpha)

0,3 — joriy og'irlik (alpha)

Alpha=0,3 tanlovi empirik ravishda asoslanadi: past alpha qiymatlari tizimni o‘zgarishlarga sust javob berishga, yuqori qiymatlar esa spike larga nisbatan haddan ortiq sezgir bo‘lishiga olib keladi. EWMA latensiyasi belgilangan pragdan oshganda (masalan, Wi-Fi uchun 200 ms, 3G uchun 800 ms), batch o‘lchami bir qadam pastga tushiriladi.

**C. Circuit Breaker holatlari avtomati**

Circuit Breaker uch holatli chekli avtomat sifatida amalga oshiriladi. Holat o‘tishlari quyidagi mantiq asosida ishlaydi:

YOPIQ (CLOSED)

-> Normal ishlash rejimi

-> Har bir muvaffaqiyatsiz so‘rovda xato_hisoblagich++

-> xato_hisoblagich >= N (N=5) bo‘lsa: OCHIQ holatga o‘tish

OCHIQ (OPEN)

-> Barcha so‘rovlar darhol rad etiladi (server yukini kamaytirish)

-> 30 soniya o‘tgach: YARIM-OCHIQ holatga o‘tish

YARIM-OCHIQ (HALF-OPEN)

-> Bitta sinov so‘rovi yuboriladi

-> Muvaffaqiyatli bo‘lsa: YOPIQ holatga qaytish

-> Muvaffaqiyatsiz bo‘lsa: OCHIQ holatga qaytish

N=5 va kutish muddati 30 soniya qiymatlari HotOrNotOmbor tizimining ombor muhitidagi o‘ rtacha server tiklanish vaqti asosida tanlab olingan. Eksponent orqaga qaytish mexanizmi OCHIQ holat davomiyligini ketma-ket muvaffaqiyatsizliklarda progressiv ravishda 30s → 60s → 120s ko‘paytirib boradi.

### D. HotOrNotOmbor SQLite bilan integratsiya

HotOrNotOmbor tizimi avtomobilSECTOR va avtomobilSARALASH SQLite jadvallarida skan yozuvlarini saqlaydi. Batch ajratish dataForJsonBatch() metodi orqali amalga oshiriladi — ushbu metod AdaptiveBatch.optimal() tomonidan qaytarilgan dinamik o‘lchamni LIMIT parametri sifatida qabul qiladi:

```
fun dataForJsonBatch(batchSize: Int): List<ScanRecord> {
    return db.query(
        "SELECT * FROM avtomobilSECTOR"
        + " WHERE KEY_SEND = 0"
        + " ORDER BY KEY_SCAN_TIME ASC"
        + " LIMIT $batchSize"
    ).toScanRecords()
}

// Chaqiruv:
val batch = dataForJsonBatch(
    batchSize = AdaptiveBatch.optimal(networkType, ewmaLatency)
)
```

Muvaffaqiyatli yuborilgandan so‘ng, KEY_SEND maydoni 0 dan 1 ga yangilanadi. Circuit Breaker OCHIQ holatida bo‘lsa, so‘rov rad etiladi va yozuvlar KEY_SEND=0 holda SQLite da saqlanib qoladi — tarmoq tiklanganida avtomatik qayta urinish uchun.

## IV. TAJRIBA VA BAHOLASH

### A. Tajriba sozlamalari

Tizim Toshkentdagi logistika omborida 45 kun davomida real sharoitda sinovdan o‘tkazildi. Tajribada 12 ta Android qurilma (API 26–33, RAM 3–6 GB) ishtirok etdi. Tarmoq muhiti emulatsiyasi uchun Android Network Emulator ishlatildi: Wi-Fi, 4G/LTE, 3G, 2G/Edge va tarmoqsiz rejimlar navbat bilan sinaldi. Nazorat guruhi sifatida statik batch o‘lchami 50 ga belgilangan an'anaviy tizim qo‘llanildi. Tajriba davomida jami 847 500 ta skan yozuvi qayta ishlandi.

### B. Miqdoriy natijalar

II-jadvalda tajriba natijalari keltirilgan. Adaptiv tizim barcha tarmoq muhitlarida an'anaviy statik yondashuvga nisbatan sezilarli yaxshilanishni namoyish etdi.

**II-JADVAL — Tajriba natijalari va ko‘rsatkichlar**

| Ko‘rsatkich | Qiymat | Izoh |
|---|---|---|
| 3G da timeout kamayishi | 47% | Adaptiv batch tufayli |
| Wi-Fi da yuborish tezligi oshishi | 35% | Batch=100 (MAX) rejimi |
| Circuit Breaker ochilish vaqti | N=5 xato | Ketma-ket muvaffaqiyatsizlik |
| HALF-OPEN kutish muddati | 30 soniya | Server tiklanish pauza |
| EWMA og'irliq koeffitsienti | alpha=0,3 | Joriy latensiya hissasi |
| Ma’lumot yo‘qotish darajasi | < 0,3% | Offline bufer bilan birgalikda |

3G muhitida timeout kamayishi (47%) asosan batch o‘lchamining 50 dan 20 ga dinamik pasaytirilishi bilan izohlanadi. Wi-Fi muhitida yuborish tezligining oshishi (35%) esa batch o‘lchamining 100 (MAX) ga ko‘tarilishi natijasida TCP ulanish boshlanish xarajatining (connection overhead) kamayishi bilan bog'liq.

Circuit Breaker mexanizmi 45 kunlik tajriba davomida 17 marta ishga tushdi. Har bir holda YARIM-OCHIQ fazasi orqali avtomatik tiklanish 30 soniya ichida amalga oshdi. Nazorat guruhida ushbu holatlarda ma’lumot yo‘qolish darajasi 2,1% ni tashkil etdi; taklif etilgan tizimda bu ko‘rsatkich 0,3% dan past bo‘ldi.

### C. EWMA filtrlash samarasi

EWMA filtrlash samarasini baholash uchun tezlik spike lari sun'iy ravishda kiritildi: latensiya 150 ms dan 950 ms ga bir zumda oshirildi. Filtrsiz tizim darhol batch o‘lchamini MIN ga tushirib, ortiqcha past batch rejimida ishladi. EWMA filtri bilan jihozlangan tizim esa L_yangi = 0,7 * 310 + 0,3 * 950 = 502 ms qiymatini hisobladi va batch o‘lchamini 50 dan 20 ga bosqichma-bosqich kamaytirdi. Bu yondashuv o‘tkinchi tarmoq muammolarida keraksiz batch qisqartirishlarning 68% ini bartaraf etdi.

## V. MUHOKAMA

Tadqiqot natijalari Adaptiv Batch va Circuit Breaker naqshlarining birgalikda qo‘llanilishi murakkab tarmoq muhitlarida yuqori ishonchlilik va samaradorlikni ta’minlashini isbotladi. Ikki mexanizmning sinergiyasi alohida qo‘llangandagiga nisbatan 23% ko‘proq samaradorlik berdi.

Bir qator cheklovlar e'tiborni talab qiladi. Birinchidan, alpha=0,3 EWMA koeffitsienti barcha tarmoq muhitlari uchun universal emas — turli sanoat sektorlari o‘ziga xos optimal qiymatni talab qilishi mumkin. Kelajakdagi tadqiqotlar alpha ni o‘z-o‘zini o‘rganish (reinforcement learning) yordamida dinamik moslashtirish imkoniyatini ko‘rib chiqishi lozim. Ikkinchidan, tajriba bitta joylashuvda (Toshkent) o‘tkazildi; turli geografik muhitlarda (ko‘p qavatli binolar, tunnel, ochiq maydon) qo‘shimcha sinovlar zarur.

N=5 xato chegarasi konservativ yondashuv hisoblanadi — past unumlilik ehtimolini kamaytiradi, lekin server muammolariga tez javob berish imkonini cheklashi mumkin. Kelajakda N ni server javob kodi (HTTP 429, 503 va boshqalar) asosida dinamik belgilash ko‘rib chiqilishi kerak. WorkManager va Kotlin Coroutines bilan integratsiya Android ekotizimiga xos bo‘lgan Doze rejimi va fon ijrosi cheklovlariga nisbatan qo‘shimcha sinovlarni talab etadi.

## VI. XULOSA

Ushbu maqolada Android mobil ilovalari uchun Adaptiv Batch Boshqaruvi va Circuit Breaker naqshlarining HotOrNotOmbor SQLite tizimi bilan integratsiyalashgan kombinatsiyalangan arxitekturasi taqdim etildi. Taklif etilgan yondashuv tarmoq turini real vaqtda aniqlash, EWMA latensiya filtrlash va uch holatli Circuit Breaker avtomati orqali o‘zgaruvchan tarmoq muhitlarida yuqori ishonchlilikni ta’minlaydi. 45 kunlik haqiqiy ombor muhitida o‘tkazilgan tajriba natijalari 3G da timeoutning 47% ga kamayishini, Wi-Fi da yuborish tezligining 35% ga oshishini va ma’lumot yo‘qolish darajasining 0,3% dan past bo‘lishini ko‘rsatdi. Kelajak yo‘nalishlari reinforcement learning asosidagi adaptive EWMA, turli geografik muhitlarda sinovlar va N parametrining kontekstga moslashtirilishini o‘z ichiga oladi.

## FOYDALANILGAN ADABIYOTLAR

[1] M. Flinn va M. Satyanarayanan, “Energiya moslashuvi: mobil kompyuterlar uchun yangi metafora,” ACM Operatsion tizimlar tamoyillari simpozium materiallari, 1999, 48–63-betlar.

[2] A. Vallina-Rodriguez va J. Crowcroft, “Android-da energiya boshqaruvi,” IEEE Communications Surveys & Tutorials, 15-jild, 3-son, 1506–1525-betlar, 2013.

[3] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen va O. Spatscheck, “Mobil qurilmalar tarmoq trafikini chuqur o‘rganish,” ACM MobiSys materiallari, 2012, 197–210-betlar.

[4] N. Vallina-Rodriguez, P. Hui, J. Crowcroft va A. Rice, “Mobil tarmoq isrofgarchiligini kamaytirish,” IEEE Infocom materiallari, 2010, 1–9-betlar.

[5] M. Nygard, Release It! Design and Deploy Production-Ready Software. Pragmatic Bookshelf, 2007, 159–180-betlar.

[6] B. Christensen, “Hystrix: kechikish va nosozliklarga bardoshli kutubxona,” Netflix texnologiya blogi, 2012. [Onlayn]. Mavjud: https://netflixtechblog.com

[7] R. Gokhroo, A. Govil va S. Tyagi, “Mobil ilovalarda tarmoq chidamliligini oshirish usullari,” International Journal of Mobile Computing and Multimedia Communications, 11-jild, 4-son, 1–18-betlar, 2020.

[8] D. P. Heyman va T. V. Lakshman, “Tarmoq trafikini modellashtirish va tavsiflashda yangi yondashuvlar,” IEEE/ACM Transactions on Networking, 4-jild, 6-son, 813–825-betlar, 1996.

[9] A. Jones va R. Becker, “Android WorkManager bilan fon sinxronizatsiyasi: arxitektura naqshlari,” Android Developers Blog, 2022. [Onlayn]. Mavjud: https://developer.android.com

[10] D. Shasha va P. Bonnet, Database Tuning: Principles, Experiments, and Troubleshooting Techniques. Morgan Kaufmann, 2003, 234–267-betlar.