Opstan (OSPT)

Whitepaper / Consensus Rules

Полная спецификация правил консенсуса, лимитов и ограничений сети (RU).

Метаданные сети

Дата документа: 2026-02-22
Источник: исходный код (архив tito.zip)
NETWORK_ID: 0x4F535054
Генезис (timestamp): 1754524800 (2025-08-07 00:00:00 UTC)
Hash заголовка генезиса:
134dc03a782a442cab7264565f3d97d3d5cf10aba2bdf516490da8140277ac52
FREEZE (MAINNET_CONSENSUS_LOCK): v2-38704739-75bd301b7482a2c1dc647cbe55
72a5b4b83c6a7a8cb2efbfd961c654ea6ece3e
Deps fingerprint (ожидаемый):
deps1-53dd0f96518e7ec25fea7161cd7dc396e5f96ffa296da3543da38a66f7a098a3
P2P protocol version: 6

Разделы Консенсус описывают валидность блоков/объектов. Политики узла — локальные ограничения mempool/сети (не консенсус).

Содержание

1. Термины и обозначения

1 OSPT = 10^18 атомов (Atoms, u128). ATOMS_PER_OSPT = 1000000000000000000. H256 — 32-байтовый sha256d-хэш. Wire V1 — бинарная сериализация ончейн-объектов.

2. Идентичность сети и генезис

NETWORK_ID: 0x4F535054. GENESIS_TIME: 1754524800 (2025-08-07 00:00:00 UTC). GENESIS_HEADER_HASH_HEX: 134dc03a782a442cab7264565f3d97d3d5cf10aba2bdf516490da8140277ac52.

2.1 FREEZE

MAINNET_CONSENSUS_LOCK должен совпасть с вычисленным отпечатком консенсуса; иначе узел не стартует.

```
v2-38704739-75bd301b7482a2c1dc647cbe5572a5b4b83c6a7a8cb2efbfd961c654ea6ece3e
```

3. Формат блока и хэширование

PoW-хэш: sha256d(encode_pow_v1(header)). encode_pow_v1 = 252 байта.
extra: 32 ASCII байта [0-9a-z] (консенсус).
ID объекта: sha256d([0x01]||canonical_encode_v1). Merkle roots пересчитываются и сверяются; дубли запрещены.

4. PoW и target

hash <= target (big-endian). target не может быть легче MIN_TARGET.

```
MIN_TARGET: 00000000ffff0000000000000000000000000000000000000000000000000000
```

Version: верхние биты по маске + version == BASE_VERSION (строго).

5. Время

MTP = медиана последних 11 блоков.

- timestamp > parent_mtp (строго).
- timestamp <= adjusted_now + 7200.

6. Ретаргет

BLOCK_TARGET_SECS=600. RETARGET_INTERVAL=144. span=86400.
На границе окна elapsed clamp: 9/10..11/10 от span.
Внутри окна: target == parent.target. На границе окна: target == ожидаемому (строго).

7. Лимиты размеров (консенсус)

Параметр	Значение
MAX_TX_BYTES	1001
MAX_MSG_BYTES	1201
MAX_PRIVMSG_BYTES	1601

MAX_SUM_TX_BYTES	4194304
MAX_SUM_MSG_BYTES	3145728
MAX_SUM_PRIVMSG_BYTES	2097152
MAX_BLOCK_TOTAL_BYTES	10485760

Общий лимит включает sum(tx)+sum(msg)+sum(priv)+252 и не должен превышать 10MB.

8. Применение блока (балансы, nonce, coinbase)

Nonce общий для msg/tx/privmsg; порядок внутри блока: msgs → transfer txs → priv_msgs. Для каждого адреса ожидается nonce = stored_nonce+1; после принятия nonce увеличивается.
Комиссии (fee) суммируются по всем объектам блока и должны совпасть с coinbase.fees.

- TransferTx: amount > 0; списывается amount+fee; получатель получает amount.
- PlainMsg/PrivMsg: списывается только fee.
- Баланс не может стать отрицательным; переполнения u128 запрещены.

CoinbaseTx (первый в txs) сверяется строго: height==header.height; miner==header.miner; reward==coinbase_reward(height); fees==∑fee.
Созревание награды: reward+fees зачисляется на высоте height+100 (COINBASE_MATURITY).

9. Ончейн-объекты и подписи

TransferTx/PlainMsg/PrivMsg подписываются Ed25519 над preimage V2, привязанным к сети:
u16_le(len(domain))||domain||NETWORK_ID_LE(4)||encode_v1(body).
Имя канала (PlainMsg): длина 1..32, только [a-z0-9].

10. Эмиссия

Фаза	Длина (блоков)	Награда (OSPT/блок)
1	52 560	240
2	210 240	120
3	210 240	60
4	210 240	30
5	210 240	15
6	210 240	8
7	210 240	4
8	210 240	2
9	∞	1

После конечных фаз: 1 OSPT/блок навсегда (лимита общей эмиссии нет).

11. Каналы: владение

Первый пост захватывает канал. Далее писать может только владелец. Смена владельца задаётся payload формата owner:<64hex>.

12. Выбор цепи (chainwork)

work = floor((2^256 - 1)/(target+1)) + 1; выбирается ветка с максимальным cumulative work.

```
work = floor((2^256 - 1) / (target + 1)) + 1
```

13. Политики узла (не консенсус)

Параметр	Значение
MEMPOOL_CAP	80000
MEMPOOL_PER_ADDR_MAX	50
MEMPOOL_TTL_BLOCKS	12

P2P Hello сверяет: protocol version=6, NETWORK_ID+genesis, consensus_lock и deps_hash.

Приложение A. Константы src/params.rs

Константа	Значение (как в исходнике)
BASE_VERSION	VERSION_TOP_BITS
BLOCK_TARGET_SECS	600
CHANNEL_NAME_MAX	32
COINBASE_MATURITY	100
DECIMALS	18
GENESIS_HEADER_HASH_HEX	"134dc03a782a442cab7264565f3d97d3d5cf10aba2bdf516490da814027 7ac52"
GENESIS_TIME	1_754_524_800
HEADER_ENCODED_LEN	252
MAX_BLOCK_TOTAL_BYTES	10 * 1024 * 1024
MAX_FUTURE_DRIFT_SECS	2 * 3600
MAX_MSG_BYTES	1201
MAX_PRIVMSG_BYTES	1601
MAX_SUM_MSG_BYTES	3 * 1024 * 1024
MAX_SUM_PRIVMSG_BYTES	2 * 1024 * 1024
MAX_SUM_TX_BYTES	4 * 1024 * 1024
MAX_TX_BYTES	1001
MEMPOOL_CAP	80_000
MEMPOOL_PER_ADDR_MAX	50
MEMPOOL_TTL_BLOCKS	12
MIN_TARGET	H256([0x00, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00,])
MIN_TARGET	см. раздел 4
MTP_WINDOW	11
NETWORK_ID	0x4F53_5054
RETARGET_INTERVAL	144
RETARGET_STRICT_STEP_MAX_DEN	10
RETARGET_STRICT_STEP_MAX_NUM	11
RETARGET_STRICT_STEP_MIN_DEN	10
RETARGET_STRICT_STEP_MIN_NUM	9
RETARGET_WINDOW_SECS	BLOCK_TARGET_SECS * RETARGET_INTERVAL
VERSION_TOP_BITS	0x20_00_00_00
VERSION_TOP_MASK	0xE0_00_00_00

Приложение B. Основные константы p2p.rs (policy)

```
const PROTOCOL_VERSION: u32 = 6;
const FRAME_HEADROOM_BYTES: usize = 256 * 1024; // enough for control messages
const MAX_WIRE_FRAME_BYTES: usize = (MAX_BLOCK_TOTAL_BYTES as usize) +
FRAME_HEADROOM_BYTES;
const PRE_HANDSHAKE_MAX_FRAME_BYTES_DEFAULT: usize = 64 * 1024;
const IO_READ_TIMEOUT: Duration = Duration::from_secs(180);
const CONNECT_RETRY_MIN: Duration = Duration::from_secs(2);
const CONNECT_RETRY_MAX: Duration = Duration::from_secs(20);
const RECONNECT_BACKOFFS_DEFAULT: [u64; 5] = [30, 60, 120, 300, 3000];
const CONNECT_TIMEOUT_DEFAULT_SECS: u64 = 8;
const TIP_INTERVAL: Duration = Duration::from_secs(10);
const PING_INTERVAL: Duration = Duration::from_secs(30);
const PEERS_GOSSIP_INTERVAL: Duration = Duration::from_secs(60);
const MAX_GOSSIP_PEERS: usize = 512;
const MAX_PEER_STR: usize = 128;
const MAX_PEERS_FROM_MSG: usize = 512;
const PEERS_FILE_MAX: usize = 10_000;
const PEERS_DELETE_ON: usize = 8_000;
const PEERS_DELETE_OFF: usize = 6_000;
const PRUNE_FAIL_THRESHOLD: u8 = 2;
const PEERSNC_FILE_MAX: usize = 100_000;
const MAX_TX_WIRE_BYTES: usize = 1024 * 1024;
const MEMPOOL_REQ_INTERVAL: Duration = Duration::from_secs(90);
const MEMPOOL_SNAPSHOT_MAX_ITEMS_DEFAULT: usize = 256;
const MEMPOOL_SNAPSHOT_MAX_BYTES_DEFAULT: usize = 256 * 1024;
const MEMPOOL_SYNC_MAX_BEHIND_DEFAULT: u64 = 2000;
const UPNP_LEASE_SECS: u32 = 3600;
const UPNP_REFRESH_EVERY: Duration = Duration::from_secs(25 * 60);
const MAX_OUTBOUND_TOTAL_DEFAULT: usize = 64;
const MAX_INBOUND_TOTAL_DEFAULT: usize = 32;
const MAX_INBOUND_PER_IP_DEFAULT: usize = 16;
const OUTBOUND_SLOT_WAIT: Duration = Duration::from_secs(1);
const OUTBOUND_TASKS_CAP_DEFAULT: usize = 256;
```