The Cell Protocol

Mutual Credit for Community Exchange:
Formal Specification and Roadmap for Connect Again

Version 1.0

R. Roberts

February 2026

Abstract

CONNECT AGAIN is a community exchange platform where neighbours help neighbours. Members earn credits by providing goods or services and spend credits by receiving them. No credits are ever created or destroyed—every exchange is a paired transfer.

This paper formalises the protocol behind CONNECT AGAIN, starting from the system as it operates today (80-member cell, invite-only admission, 5-credit starting balance, 10-credit debt floor) and extending toward the full Cell protocol: household accounts, cell accounts for shared resources, inter-cell federation with contagion bounds, and automatic emergency controls.

Each section connects formal properties to concrete app mechanics. The paper serves both as a specification for the current system and as a roadmap for where it goes next.

Contents

1 Introduction **3**
- 1.1 What Connect Again Is . . . 3
- 1.2 Why Formalise It . . . 3
- 1.3 Design Principles . . . 3

2 The System Today **3**
- 2.1 Parameters . . . 4
- 2.2 The Exchange Flow . . . 4
- 2.3 Admission: The Invite Chain . . . 4
- 2.4 What the Starting Balance Means . . . 5

3 Formal Model **5**
- 3.1 Entities . . . 5
- 3.2 Core Invariants . . . 5

4 Bounded Extraction: Why the Floor Matters **6**
- 4.1 Multi-Identity Attacks (Sybil) . . . 6

5 The Cooperation Condition: Why People Stay Honest **6**
- 5.1 The Model . . . 6
- 5.2 What This Means in Practice . . . 7

6 Roadmap: Cell Accounts **7**

7 Roadmap: Slow Privilege **8**

8 Roadmap: Household Accounts 8

9 Roadmap: Federation 9
9.1 How It Works . . . 9
9.2 Exposure Caps . . . 9
9.3 Severability . . . 10

10 Roadmap: Emergency Mode 10
10.1 Risk States . . . 10
10.2 Escalation Is Automatic . . . 10
10.3 De-Escalation Is Evidence-Based . . . 10

11 Security Summary 10
11.1 Threat Model . . . 10
11.2 What Governance Cannot Do . . . 11
11.3 Why This Is Not a Currency . . . 11

12 Implementation Roadmap 11
12.1 What Exists Today (Level 1) . . . 11
12.2 Phase 1: Cell Accounts . . . 11
12.3 Phase 2: Slow Privilege . . . 12
12.4 Phase 3: Household Accounts . . . 12
12.5 Phase 4: Commitments and Scheduling . . . 12
12.6 Phase 5: Emergency Mode . . . 12
12.7 Phase 6: Federation . . . 12

13 Conclusion 12

A Notation Reference 12

B Formal Proofs 12
B.1 Conservation Under Transactions . . . 12
B.2 Conservation Under Cell Account Operations . . . 13
B.3 Bounded Extraction . . . 13
B.4 Time-Dependent Extraction Bound . . . 13
B.5 Compound Severance Bound . . . 13

1 Introduction

1.1 What Connect Again Is

Connect Again is a community exchange platform, currently operating as a single cell of up to 80 members. Members post offers ("I can do X") and requests ("I need Y"), then arrange exchanges directly. Credits transfer from receiver to provider upon completion.

The system is:

- **Invite-only**: each member starts with 2 invites, earns +1 per 3 completed exchanges, up to a maximum of 5. This limits growth to trusted chains and prevents anyone from creating accounts in bulk.
- **Credit-conserving**: every exchange subtracts credits from the receiver and adds the same amount to the provider. No credits are printed or destroyed.
- **Debt-bounded**: no member can go below -10 credits. This caps the damage any single bad actor can do.
- **Local-first**: there is no central bank, no interest, no conversion to cash. Credits are internal bookkeeping only.

1.2 Why Formalise It

A community exchange platform is only as trustworthy as its rules. If the rules have gaps, bad actors exploit them. If the rules are unwritten, disputes have no resolution.

This paper does three things:

1. **Specifies what the system does today**, with precise definitions and proofs that the safety properties hold.
2. **Identifies what's missing**: household accounts, shared funds, federation, emergency controls.
3. **Maps the path from here to there**, so that each upgrade preserves the properties that make the system safe.

1.3 Design Principles

These apply to every decision, current and future:

1. **No credit creation.** Every credit that exists was transferred from somewhere.
2. **Bounded loss.** The worst any identity can do is limited by a hard floor.
3. **Admission friction.** Growing the network is deliberately slow.
4. **No yield, no speculation.** Credits are not an investment. There is no interest, no staking, no convertibility to cash.
5. **Local governance.** Disputes are resolved by the community, not an algorithm.

2 The System Today

This section describes Connect Again as it operates now, with the formal properties stated alongside the implementation.

Table 1: Current Connect Again parameters.

Parameter	Value	Implementation
Member cap (N)	80	`MEMBER_CAP` in `supabase.js`
Starting balance (b_0)	5 credits	Default from `getBalance()`
Debt floor	-10	`complete_exchange` RPC check
Credit limit (L)	10	Implicit: floor at -10, so $L = 10$
Exchange range	1–10 credits	UI constraint on proposal form
Base invites	2	`BASE_INVITE_LIMIT`
Max invites	5	`MAX_INVITE_LIMIT`
Invite earn rate	+1 per 3 exchanges	`getInviteAllowance()`

2.1 Parameters

2.2 The Exchange Flow

1. **Propose.** Member A sees a listing by Member B and proposes an exchange, specifying the credit amount (1–10). A balance warning appears if A is the receiver and the amount would push them below -10.

2. **Accept.** Member B reviews and accepts the proposal.

3. **Confirm (both sides).** After the real-world exchange happens, both members confirm. Once one side confirms, *neither side can cancel*—this protects the provider who has already done the work.

4. **Complete.** When both confirmations are in, the `complete_exchange` database function executes atomically:

$$b_{\text{receiver}} \leftarrow b_{\text{receiver}} - c, \qquad b_{\text{provider}} \leftarrow b_{\text{provider}} + c$$

   subject to $b_{\text{receiver}} - c \geq -10$.

2.3 Admission: The Invite Chain

Sybil resistance (preventing fake accounts) is the hardest problem in any open system. Connect Again uses a social mechanism:

- Each new member must be invited by an existing member.
- Each member starts with 2 invites.
- You earn +1 invite for every 3 completed exchanges, up to 5 total.
- New members go through a profile review before activation.

This means:

- Nobody can create accounts in bulk—you need a real member to vouch for each one.
- Invites are earned through participation, not granted freely.
- The invite chain creates accountability: if someone you invited causes problems, it reflects on you.

In protocol terms, this is *admission friction*—the key defence against multi-identity attacks (Section 4.1).

2.4 What the Starting Balance Means

Every new member starts with 5 credits. In a pure zero-sum system, all balances would begin at zero and the total sum would always be zero. With a 5-credit welcome grant, the total sum across all members is $5N$ (currently up to 400 credits for 80 members).

This does *not* violate the conservation principle, provided we understand it correctly:

Invariant 2.1 (Conservation — I1)**.** *No operation other than admission changes the total balance. Every exchange subtracts from one member and adds the same amount to another. The total is constant between admissions.*

The starting balance is a one-time grant that reduces friction for new members (they can receive services immediately without first needing to earn). In the full protocol (Section 6), this grant comes from a *welcome account*—a cell-level fund that absorbs the corresponding deficit, making the total sum exactly zero. The current implementation skips this intermediary for simplicity.

3 Formal Model

We now state the formal properties that make the system safe, starting from the simple individual-account model that Connect Again uses today.

3.1 Entities

Definition 3.1 (Cell)**.** A cell $\mathcal{C}$ at time t consists of a set of members $M(t) = \{1, \ldots, N(t)\}$ where $N(t) \leq N_{\max} = 80$. Each member i has:

- Balance $b_i(t) \in \mathbb{R}$.
- Credit limit $L_i > 0$ (currently $L = 10$ for all members).
- Reserve $r_i(t) \geq 0$ (not yet implemented; for future escrowed commitments).

Definition 3.2 (Transaction)**.** A transaction $T(i, j, v)$ with $v > 0$ (receiver i, provider j):

$$b_i \leftarrow b_i - v, \qquad b_j \leftarrow b_j + v$$

subject to feasibility: $b_i - v \geq -L_i$.

3.2 Core Invariants

Invariant 3.1 (Hard Debt Floor — I2)**.**

$$b_i(t) \geq -L_i \qquad \forall\, i,\, t$$

Currently: $b_i(t) \geq -10$ *for all members at all times. This is enforced by the* `complete_exchange` *RPC, which rejects the transaction if it would push the receiver below* -10.

These two invariants—conservation and bounded debt—are the foundation everything else builds on.

4 Bounded Extraction: Why the Floor Matters

The debt floor isn't just a convenience feature. It's the fundamental security property of the system.

Theorem 4.1 (Bounded Individual Extraction)**.** *Let member i have credit limit L_i and current balance b_i. The maximum value i can extract (receive without giving back) is:*

$$G_i \leq b_i + L_i$$

Proof. Each unit received reduces b_i by one. The floor at $-L_i$ blocks further receiving. Maximum extraction from b_i to $-L_i$ is the distance $b_i + L_i$. □

Example 4.1 (Current System)**.** A new member starts with $b = 5$ and $L = 10$. Maximum one-shot extraction: $5 + 10 = 15$ credits. That's 15 hours of community labour. After extracting, they sit at -10 and cannot receive any more services until they give back.

A bad actor who joins, takes 15 credits worth of services, and disappears has cost the community 15 labour-hours. That's the security budget per identity.

4.1 Multi-Identity Attacks (Sybil)

Theorem 4.2 (Bounded Multi-Identity Extraction)**.** *If an attacker controls S admitted identities, total extraction is:*

$$G_{total} \leq \sum_{s=1}^{S}(b_s + L_s) = S \cdot (5 + 10) = 15S$$

This is why admission friction matters. With 2 invites per member and the invite-earn mechanism, getting 3 fake identities admitted requires at least 2 real members willing to vouch (or one member who completes 3 exchanges to earn a third invite). The practical S is small.

Example 4.2 (Attack Cost)**.** An attacker who somehow gets $S = 3$ identities admitted can extract at most $3 \times 15 = 45$ credits. That's 45 labour-hours across an 80-person community—painful but survivable. And it required finding 3 people willing to use invites on fake identities, plus passing profile review 3 times.

5 The Cooperation Condition: Why People Stay Honest

The debt floor limits what a bad actor can *take*. But why do people *stay*? We need cooperation to be the rational choice.

5.1 The Model

Each period, a cooperating member gets utility u^C from the community (services received, social value, security of mutual support). A defector extracts αL worth of goods and leaves, gaining outside-option utility w thereafter.

We separate two factors that affect long-term thinking:

- $\rho \in (0, 1)$: **time preference**—how much you value tomorrow vs. today.
- $\pi(t) \in (0, 1)$: **continuation probability**—how likely it is the community persists to the next period.

Theorem 5.1 (Cooperation Condition)**.** *Cooperation is individually rational when:*

$$\frac{u^C}{1-\rho\pi} \geq \alpha L + w$$

Equivalently, the credit limit must satisfy:

$$L \leq \frac{u^C - (1-\rho\pi)\cdot w}{\alpha \cdot (1-\rho\pi)} \tag{1}$$

5.2 What This Means in Practice

When things are going well (π high): people value the community, the temptation to defect is low, and even a generous credit limit is safe.

When the community feels shaky (π drops): the future seems uncertain, so the temptation to grab-and-go increases. The permissible L tightens. This is why emergency controls (Section 10) reduce credit limits when stress indicators rise.

When outside options improve (w rises): if people can get equivalent services elsewhere, cooperation becomes harder to sustain. This is normal—mutual credit works best where alternatives are limited or where the community provides something money can't buy (trust, belonging, local knowledge).

Example 5.1 (Current Parameters)**.** With $\rho = 0.95$, $u^C = 8$ (modest community value), $w = 1$, $\alpha = 1$, $\pi = 0.90$:

$$L \leq \frac{8 - 0.145 \times 1}{1 \times 0.145} = \frac{7.855}{0.145} \approx 54$$

Our current $L = 10$ is well within the safe zone. Even if π dropped to 0.50 (members think there's a coin-flip chance the community survives):

$$L \leq \frac{8 - 0.525 \times 1}{0.525} \approx 14$$

Still above our $L = 10$. The system has substantial headroom.

6 Roadmap: Cell Accounts

Status: Not yet implemented. This section specifies the next major upgrade.

As the community grows, it needs shared funds: a welcome account for new-member grants, a meal fund for communal cooking, a maintenance fund for repairs. These need formal treatment to ensure they can't break conservation.

Definition 6.1 (Cell Account)**.** A cell account a_p has:

- Balance $b_{a_p}(t) \in \mathbb{R}$.
- Floor limit: $b_{a_p}(t) \geq -L_{a_p}$.
- Ceiling limit: $b_{a_p}(t) \leq U_{a_p}$.

Cell accounts are **not members**. They cannot propose exchanges, vote, or hold invites. They are passive ledger entries that receive and disburse credits under governance rules.

Key rule. Cell accounts participate in the conservation sum. Every operation involving a cell account is a paired transfer:

$$b_h \leftarrow b_h - v, \quad b_{a_p} \leftarrow b_{a_p} + v \qquad \text{or vice versa}$$

Proposition 6.1 (Conservation Under Cell Accounts)**.** *If every cell account operation is a paired transfer, then the total balance (members + cell accounts) is constant.*

The proof is immediate: $-v + v = 0$.

Welcome Account. Currently, the 5-credit starting balance is created from nothing. With a welcome account, the grant comes *from* the welcome account:

$$b_{\text{welcome}} \leftarrow b_{\text{welcome}} - 5$$
$$b_{\text{new member}} \leftarrow 0 + 5 = 5$$

The welcome account goes negative (funded by small periodic levies from all members), and the total sum stays at zero. This is cleaner and eliminates the "credits from nowhere" edge case.

Meal Fund, Maintenance Fund, Tool Library. Same pattern: funded by levies, disbursed under rules, bounded by floor and ceiling limits. An attacker who captures governance can drain a cell account, but only to its floor—a bounded, auditable loss.

7 Roadmap: Slow Privilege

Status: Partially implemented (newcomers start with 5 credits and earn invites gradually). Formal limit growth not yet implemented.

Currently all members have the same credit limit ($L = 10$). The full protocol introduces *slow privilege*: newcomers start with a low limit that grows over time.

Proposition 7.1 (Time-Dependent Extraction Bound)**.** *If credit limits grow at most η per period: $|L_i(t+1) - L_i(t)| \leq \eta$, then maximum extraction over T periods is:*

$$G_i(T) \leq b_i(t_0) + L_i(t_0) + \eta \cdot T$$

Slow privilege means an attacker must cooperate for a long time to build up extraction capacity, increasing the chance of detection.

Example 7.1 (Future Parameters)**.** If we increase the full limit to $L = 20$ and start newcomers at $L_{\text{new}} = 5$ with $\eta = 0.5$ credits/week:

- 30 weeks to reach full limit.
- 3 Sybil identities over 26 weeks: max extraction $3 \times (5 + 0.5 \times 26) = 54$ credits.
- That's 54 labour-hours—survivable for an 80-person community, and it required 6 months of fake cooperation.

8 Roadmap: Household Accounts

Status: Not yet implemented.

Working adults naturally run surpluses; dependents (children, elderly, disabled) run deficits. Without household accounts, dependents hit the floor quickly.

The household model groups family members under a joint balance:

- Joint balance b_h shared across household members.
- Individual profiles for reputation and task assignment.
- Working adults' surpluses naturally carry dependents.

Household credit limit accounts for composition:

$$L_h = n_h \cdot L_{\text{adult}} + \min(d_h,\, d_{\max}) \cdot L_{\text{dep}}$$

with $L_{\text{dep}} \ll L_{\text{adult}}$. This lets families participate without giving dependents independent extraction capacity.

9 Roadmap: Federation

Status: Not yet implemented. Currently Connect Again *operates as a single cell. Federation is needed when multiple cells exist.*

A single cell may lack specialisation or redundancy. Federation enables limited trade between cells. But it's also the primary route to systemic failure: if cells can run persistent net deficits with others, the network recreates the leverage dynamics of the financial system.

9.1 How It Works

Each cell maintains its own internal ledger. Inter-cell trade goes through clearing accounts:

1. In the buying cell: buyer pays the clearing account.
2. In the selling cell: clearing account pays the seller.
3. Federation positions track the net balance between cells.

Each cell's internal conservation is preserved. The federation layer is a separate netting mechanism.

9.2 Exposure Caps

Definition 9.1 (Aggregate Credit Capacity)**.** $\Lambda_k = \sum_{i \in M_k} L_i$. For Connect Again today: $\Lambda = 80 \times 10 = 800$ credits.

Constraint 9.1 (Aggregate Absolute Exposure — F1)**.**

$$\sum_{l \in \text{links}(k)} |B_{k,l}(t)| \leq \beta \cdot \Lambda_k(t)$$

where $\beta \in [0.05, 0.10]$.

The key insight: we bound the *sum of absolute* per-link positions, not just the net aggregate. This prevents a cell from having large offsetting positions that look safe but become dangerous if one link is severed.

Theorem 9.1 (Compound Severance Bound)**.** *Under F1, if any subset of federation links is simultaneously severed, total value at risk is bounded by* $\beta \cdot \Lambda_k$*.*

Example 9.1 (Future Federation)**.** With $\beta = 0.05$ and $\Lambda = 800$: max federation exposure $= 0.05 \times 800 = 40$ credits. Even if all federation links collapse simultaneously, the cell loses at most 40 credits—about 2 person-weeks of labour. Survivable.

9.3 Severability

Any cell can be isolated from the federation without breaking its internal accounting. Federation connections can be cut—the internal ledger keeps working. This is the "no contagion" property.

10 Roadmap: Emergency Mode

Status: Not yet implemented.

Conditions change. If the community comes under stress (members leaving, disputes rising, participation dropping), the protocol should tighten automatically.

10.1 Risk States

$$\textbf{Normal} \xrightarrow{\text{stress rises}} \textbf{Stressed} \xrightarrow{\text{stress rises}} \textbf{Panic}$$

Stressed. Reduce newcomer limits, require stronger admission vetting, escrow essential commitments.

Panic. Tighten all limits ($L_i \leftarrow \lambda L_i$, $\lambda < 1$), freeze federation, prioritise essential task matching.

10.2 Escalation Is Automatic

If stress indicators exceed thresholds, the system escalates *regardless of governance preference.* A council cannot vote to suppress an emergency.

10.3 De-Escalation Is Evidence-Based

Without de-escalation rules, a cell that enters Panic stays there forever. De-escalation requires:

- Stress indicators below the threshold for multiple consecutive periods (cooldown).
- Gradual parameter restoration (no instant snapback).
- Hysteresis gap to prevent flapping between states.

The asymmetry is deliberate: escalation is automatic (safety); de-escalation requires evidence (caution).

Proposition 10.1 (De-Escalation Preserves Safety)**.** *Gradual restoration does not weaken bounded loss. At every point during restoration, conservation holds (no balances change), and the debt floor is maintained (limits increase, which relaxes the floor). De-escalation restores the normal security posture; it does not create a new vulnerability.*

11 Security Summary

11.1 Threat Model

Opportunistic defector: joins, takes services, disappears. *Defence*: debt floor bounds extraction at $b_i + L_i$.

Sybil attacker: creates multiple identities to multiply extraction. *Defence*: invite-only admission limits S. With 2 invites per member and earn-through-exchange, bulk account creation is impractical.

Colluding group: members coordinate to inflate each other's standing or drain resources. *Defence*: all transactions are paired transfers; you can't create net value through collusion. Governance capture is limited because governance cannot violate conservation.

Free rider: receives without giving. *Defence*: they hit the floor at $-L$ and can't receive further. The system is self-regulating: free riders are naturally throttled.

11.2 What Governance Cannot Do

The conservation invariant is the structural defence. No governance action—however well-intentioned or corrupt—can:

- Create credits from nothing (every credit requires a counterparty).
- Remove the debt floor (it's enforced at the database level).
- Grant unlimited invites (the system caps them).

These are not policy choices. They are hard constraints in the code.

11.3 Why This Is Not a Currency

Connect Again avoids structures that invite regulatory attention or capture:

1. **No issuer.** There is no entity that "issues" credits. Balances are net positions.
2. **No yield.** No interest, no staking, no return on holding credits.
3. **No convertibility.** Credits cannot be redeemed for cash.
4. **No global network.** Each cell is independent. There is no global ledger to capture or regulate.

12 Implementation Roadmap

12.1 What Exists Today (Level 1)

Table 2: Current implementation status.

Feature	Status	Notes
Individual ledger	Complete	Conservation enforced
Debt floor (-10)	Complete	Database-level check
Invite-only admission	Complete	2 base, earn up to 5
Profile review	Complete	PENDING → REVIEW → ACCEPTED
Exchange flow	Complete	Propose → accept → confirm → complete
Balance warnings	Complete	Warns proposer if near floor
Cancel protection	Complete	No cancel after partial confirmation
Listings (offers/requests)	Complete	Categories, areas, skill tags
Member directory	Complete	Profiles, skills, areas
Feedback system	Complete	In-app feedback form

12.2 Phase 1: Cell Accounts

Add welcome account, meal fund, maintenance fund. Make the starting-credit grant come from the welcome account. Enforce floor and ceiling limits. Preserve conservation with formal proofs.

12.3 Phase 2: Slow Privilege

Newcomers start with $L_{\text{new}} = 5$, growing at $\eta = 0.5$ per week. Rate-limit all credit limit changes. This strengthens Sybil resistance from "bounded by admission friction" to "bounded by admission friction AND time."

12.4 Phase 3: Household Accounts

Joint balances for families. Dependent support without independent extraction capacity. Household credit limits based on composition.

12.5 Phase 4: Commitments and Scheduling

Escrowed commitments for recurring obligations (meal rotas, maintenance shifts). This is the bridge from "exchange platform" to "coordination platform"—the system that lets a community plan, not just trade.

12.6 Phase 5: Emergency Mode

Stress indicators, automatic escalation, evidence-based de-escalation. This makes the system adaptive: it tightens when things get rough and relaxes when conditions improve.

12.7 Phase 6: Federation

Bilateral links between cells with aggregate absolute exposure caps. Severability. Automatic quarantine. Only needed when multiple cells exist.

13 Conclusion

Connect Again is a living system, not a theoretical exercise. The protocol it runs on enforces:

- **Conservation**: no credits are created or destroyed.
- **Bounded loss**: the worst any identity can do is capped at $b + L = 15$ credits.
- **Admission friction**: the invite chain prevents bulk account creation.
- **No capture points**: no issuer, no yield, no global ledger, no money printer.

The roadmap adds household accounts, cell accounts, slow privilege, commitments, emergency controls, and federation—each preserving these core properties while expanding what the community can do.

The math isn't decoration. It's the guarantee that the system does what it says, that bad actors are bounded, and that each upgrade doesn't break what came before.

A Notation Reference

B Formal Proofs

B.1 Conservation Under Transactions

Proof of I1. A transaction $T(i, j, v)$ updates: $b_i \leftarrow b_i - v$, $b_j \leftarrow b_j + v$. Change in total: $-v + v = 0$. No other operation modifies balances (admission grants are modelled as cell account transfers in the full protocol). $\square$

Table 3: Symbol definitions.

Symbol	Meaning
$\mathcal{C}_k$	Cell k
$M(t)$	Set of members at time t
N	Number of members
$b_i(t)$	Balance of member i at time t
L_i	Credit limit for member i
$r_i(t)$	Reserve (escrowed) for member i
$b_{a_p}(t)$	Balance of cell account a_p
L_{a_p}	Floor limit for cell account a_p
U_{a_p}	Ceiling limit for cell account a_p
G_i	Maximum extraction by member i
η	Rate limit on credit limit changes
ρ	Time discount factor
$\pi(t)$	Continuation probability
u^C	Per-period utility from cooperation
α	Conversion factor (credits to utility)
w	Outside option utility
$B_{k,l}(t)$	Net position of cell k on federation link l
Λ_k	Aggregate credit capacity of cell k
β	Federation exposure cap
$\sigma(t)$	Combined stress index

B.2 Conservation Under Cell Account Operations

Proof. Household $\to$ cell account: b_h decreases by v, b_{a_p} increases by v. Net change: 0. Cell account $\to$ household: symmetric. No other operation type is admissible. $\square$

B.3 Bounded Extraction

Proof of Theorem 4.1. From balance b_i, each unit consumed reduces b_i by one. Floor at $-L_i$ blocks further spending. Distance: $b_i - (-L_i) = b_i + L_i$. $\square$

B.4 Time-Dependent Extraction Bound

Proof of Proposition 7.1. Rate limit: $L_i(t) \leq L_i(t_0) + \eta(t - t_0)$. Attacker cooperates for T periods, then extracts at max limit. Worst case: $G_i = b_i(t_0) + L_i(t_0) + \eta T$. $\square$

B.5 Compound Severance Bound

Proof of Theorem 9.1. Upon severance of link l, position $B_{k,l}$ freezes. Total at-risk value: $\sum_{l \in S} |B_{k,l}| \leq \sum_l |B_{k,l}| \leq \beta\Lambda_k$ by F1. Holds regardless of which or how many links are severed. $\square$