FastChain Core Optimization Problem. "F"

@Cryptis - Cryptix-Network.org

December 2025

1 On the objective function F

In optimization notation, the symbol F commonly denotes the *objective function* (also called the cost or loss when minimization is intended). An optimization problem is typically written as

$$\min_{x} \; F(x) \qquad subjectto \qquad g_i(x) \le 0, \; h_j(x) = 0,$$

where x collects decision variables and the constraints g_i, h_j enforce physical, resource, or operational limits.

1.1 Why the original document has no explicit F

The FastChain draft is written primarily in control-theoretic language (adaptive loops, Lyapunov function candidate V, and heuristic reward $R(\lambda)$). It therefore describes objectives and performance metrics (latency, fork rate, mempool growth, throughput) without aggregating them into a single scalar objective F. This is why an explicit F is *not present* in the original text.

1.2 A natural candidate objective for FastChain

To convert the control description into a clean, mathematically complete constrained optimization problem, we introduce a scalar objective F that penalizes deviations of key performance metrics from their desired targets. A straightforward choice is a weighted-sum quadratic cost:

$$\begin{aligned}&\min_{\lambda,\, F_{fan}} \; F(\lambda, F_{fan}) = a_1 \big(\mathrm{hdr}_{p90}(\lambda, F_{fan}) - \mathrm{hdr}_{target}\big)^2 \\ &+ a_2 \big(\mathrm{fork}(\lambda, F_{fan}) - \mathrm{fork}_{target}\big)^2 \\ &+ a_3 \big(\mathrm{mempool_growth}(\lambda, F_{fan})\big)^2\end{aligned} \tag{1}$$

Here:

- Decision variables: λ (global microblock generation rate, blocks/s) and F_{fan} (fanout / gossip parameter). Other decision variables (e.g., adaptive gains α, κ) can be included if desired.

- hdr_{p90} is the observed 90th-percentile header propagation latency (seconds).
- fork is the observed fork probability (or a fork-rate metric).
- mempool_growth is the observed mempool growth rate (tx/s) — positive values indicate backlog growth.
- $\text{hdr}_{target}, \text{fork}_{target}$ are desired target values (design parameters).
- $a_1, a_2, a_3 \geq 0$ are scalar weights that trade off the three objectives.

This objective F is intentionally modular: additional terms (e.g., penalties on bandwidth use, rewards for throughput, or regularizers on control changes) can be added:

$$F(\cdot) \leftarrow F(\cdot) + a_4\big(\text{bandwidth_usage}(\lambda, F_{fan}) - B_{target}\big)^2 + a_5\,\Delta\lambda^2,$$

where $\Delta\lambda$ penalizes rapid changes in block rate.

1.3 Constraints

The optimization must respect physical and operational constraints. Representative constraints are:

Bandwidth (per-node propagation capacity)

$$\lambda \leq \lambda_{\max}^{bandwidth} = \frac{U \cdot e}{B_s \cdot k},$$

where U is upload bandwidth (Bytes/s), B_s microblock header size (Bytes), k average parents, and e efficiency factor.

CPU / verification limit

$$\lambda \leq \lambda_{\max}^{CPU},$$

set by the maximum signature verification rate.

Mempool stability

$$\mu_{tx} \leq \lambda \cdot n_{tx},$$

i.e., arrival rate must not exceed processing capacity (average transactions per block n_{tx}).

Fork-probability bound Using a Poisson approximation:

$$P_{fork}(\lambda) = 1 - e^{-\lambda\Delta} \leq P_{\max},$$

where Δ is characteristic propagation latency and $P_{\max}$ is an acceptable fork probability threshold.

Fanout bounds

$$F_{min} \leq F_{fan} \leq F_{max}.$$

1.4 Interpretation and choices

- **Why quadratic terms?** Quadratic penalties are smooth, convex (in the simplest approximations) and penalize large deviations heavily. They are convenient for both analytical reasoning and numerical optimization. Alternatives include absolute-value penalties (robust to outliers) or asymmetric penalties if over- and under-shooting have different costs.

- **Choosing weights a_i.** Weights express designer priorities: e.g., if finality/fork avoidance is critical, increase a_2. Weights can be tuned via grid search or by solving a multi-objective Pareto analysis.

- **Stochasticity.** Metrics are stochastic (latencies, arrivals, packet loss). The optimization can be cast as a stochastic optimization or Model Predictive Control (MPC) problem that re-solves periodically using observed statistics (EMA estimates in the paper can provide these statistics).

- **Relation to Lyapunov V.** The Lyapunov candidate V in the paper is conceptually similar to an objective: both measure deviation from desired operating points. V is used to reason about closed-loop stability, while F above is an explicit optimization objective used for parameter selection or as a cost in MPC.

1.5 Alternative formulations

1. **Maximize throughput subject to stability:**

$$\max_{\lambda, F_{fan}} \ \lambda \cdot n_{tx} \quad s.t. constraints and \text{fork}(\lambda, F_{fan}) \leq \text{fork}_{target},\ \text{hdr}_{p90} \leq \text{hdr}_{target}.$$

2. **Multi-objective form:** do not aggregate into a single F; instead, treat $(\text{hdr}_{p90}, \text{fork}, \text{mempool_growth})$ as a vector objective and compute Pareto front / trade-offs.

3. **MPC / dynamic control:** minimize expected cumulative cost over horizon T:

$$\min_{\{\lambda_t, F_{fan,t}\}_{t=0}^{T-1}} \ E\Big[\sum_{t=0}^{T-1} F(\lambda_t, F_{fan,t})\Big]$$

subject to dynamic constraints and system evolution.