

(a) シーケンス図を図1に示す。

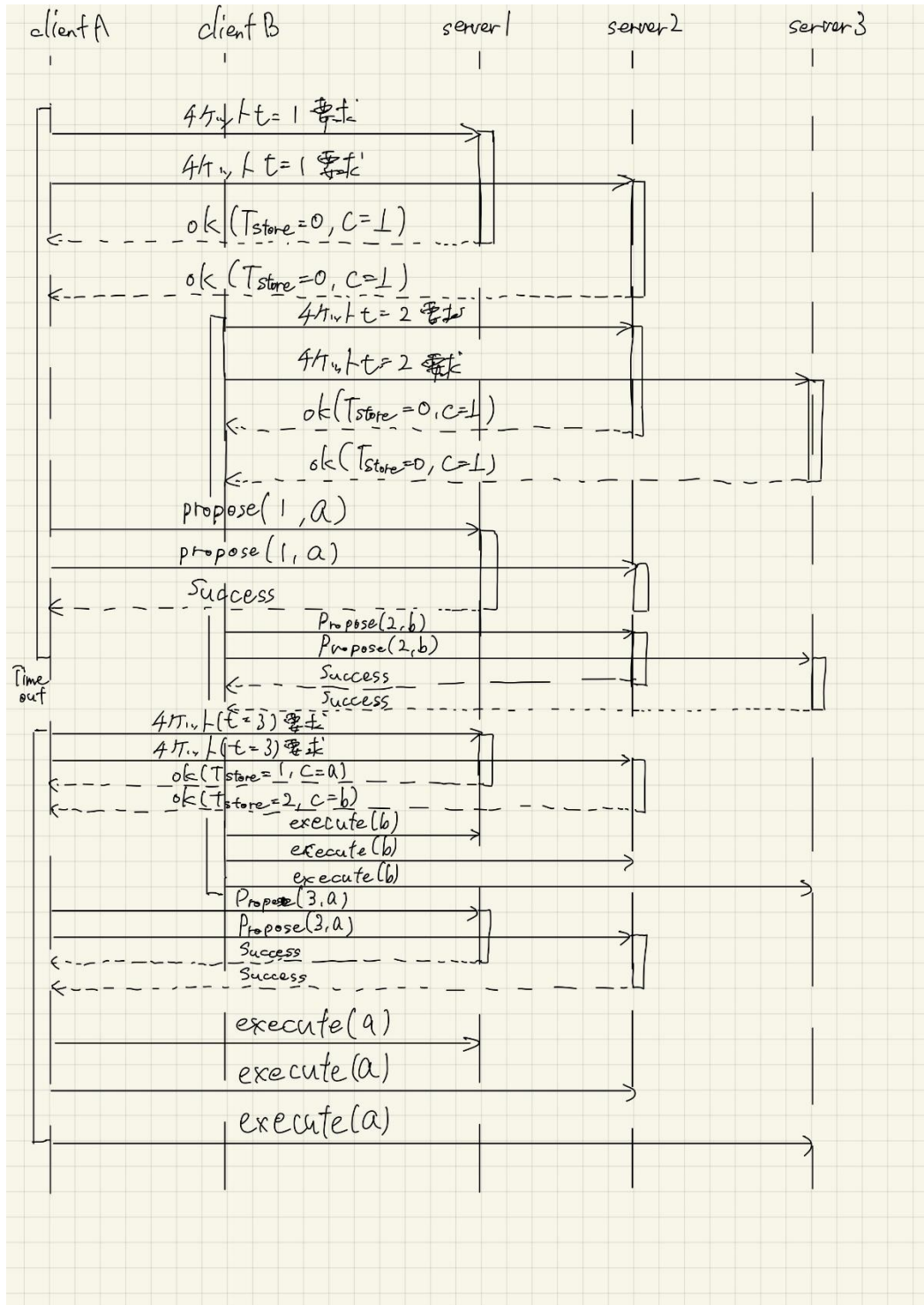


図1 シーケンス図(a)

(b) シーケンス図を図2に示す。

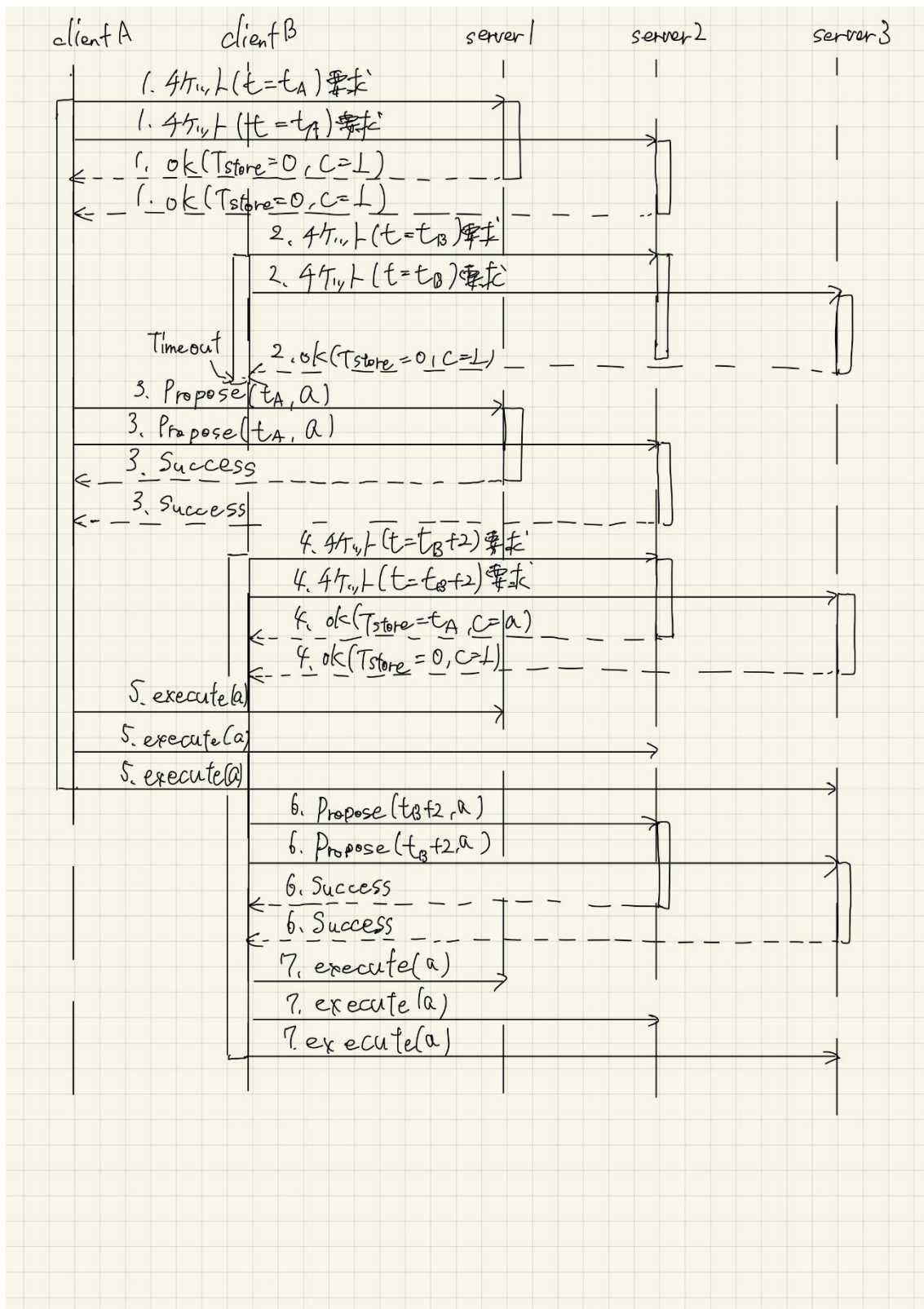


図2 シーケンス図2(b)

以下、この図に従って動作の説明を行う。

1. クライアント A がチケット $t = t_A$ をサーバ 1,2 に要求する。サーバ 1,2 では $t = t_A > 0 = T_{max}$ であるから、 $T_{max} = t_A$ と更新し、 $ok(T_{store} = 0, C = \perp)$ を返す。
2. クライアント B がチケット $t = t_B$ をサーバ 2,3 に要求する。サーバ 3 では $t = t_B > 0 = T_{max}$ であるから、 $T_{max} = t_B$ と更新し、 $ok(T_{store} = 0, C = \perp)$ を返す。サーバ 2 では $t = t_A \leq t_B = T_{max}$ であるから、何も返さない。
3. 1. で過半数のサーバが ok を送信しており、 T_{store} の最大値は $T_{store} = 0 \leq 0$ であるからコマンド c は更新せず、 $propose(t = t_A, c = a)$ をサーバ 1,2 に送信する。 $t = t_A = T_{max}$ であるからサーバ 1,2 は $C = a, T_{store} = t_A$ と更新し、 $success$ と送信する。
4. δ_B 秒だけ待機してもサーバ 2 からの返答がないため、 $t = t_B + 2$ と更新し、Phase1 の処理をやり直す。
クライアント B がチケット $t = t_B + 2$ をサーバ 2,3 に要求する。サーバ 2,3 では $t = t_B + 2 > t_A = t_B = T_{max}$ であるから、 $T_{max} = t_B + 2$ と更新し、サーバ 1 は $ok(T_{store} = t_A, c = a)$ を、サーバ 2 は $ok(T_{store} = 0, c = \perp)$ を返す。
5. 過半数が $success$ を応答しているため、クライアント A が $execute(c = a)$ を全サーバに送信する。
6. 4. で過半数のサーバが ok を送信しており、 T_{store} の最大値は $T_{store} = t_A > 0$ であるから、クライアント 2 は T_{store} に対応するコマンド a に対し、 $c = a$ と更新し、 $propose(t = t_B + 2, c = a)$ をサーバ 2,3 に送信する。 $t = t_B + 2 = T_{max}$ であるから、サーバ 2,3 は $C = a, T_{store} = t_B + 2$ と更新し、 $success$ と送信する。
7. 過半数が $success$ と応答しているため、クライアント B が $execute(c = a)$ を全サーバに送信する。

以上で動作が完了する。