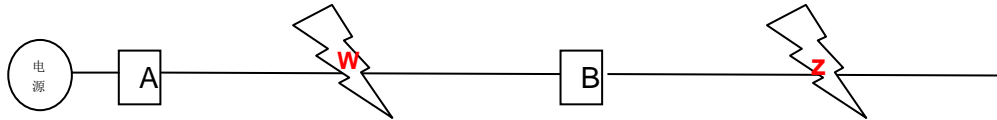


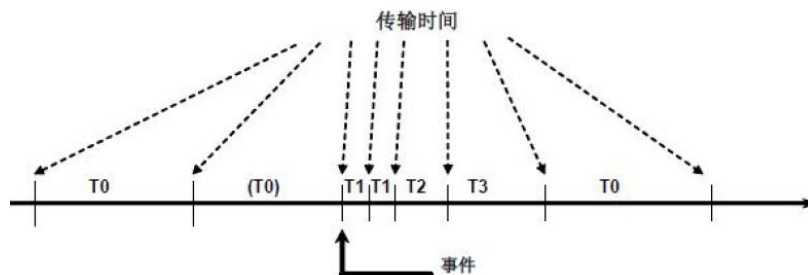
用 IEC61850 GOOSE 实现电流闭锁式差动保护的方法



A. IEC61850 GOOSE 原理实现差动保护的基本原理

- 1) A 和 B 两侧过流保护内设置一个电流阈值，当电流大于该阈值就认为有线路有故障，生成“启动信号”。（阈值一般要大于 2 倍负荷电流，该方式会造成故障检测的灵敏度降低，无法看见高阻故障）。
- 2) 过电流判据算法时间通常为 20ms，这样构建的差动保护系统的启动时间不可能大于 20ms。
- 3) “启动信号”用 GOOSE 机制从装置的以太网口发出，通过以太网交换机传到对侧保护装置。
- 4) 当 A 装置未收到 B 发来的“启动信号”，B 装置在线，且 A 装置有“启动信号”，认为线路有区内故障，A 装置跳闸。

B. GOOSE 原理



- T0: 稳定条件(长时间无事件)下重传
- (T0): 稳定条件下的重传可能被事件缩短
- T1: 事件发生后,最短的传输时间
- T2, T3: 直到获得稳定条件的重传时间

- GOOSE 采用广播方式单相发送信息，这意味着没有确认信号，所以必须使用重复发送机制确保信息传送。发送时间间隔的计算公式为： $t_s = 2^N * S$ ($N=0,1,2,3...for t_s < Z, S = 1 ms$), $T1=1ms, T2=2ms, T3=4ms....$

T0 一般定义为 500ms，为心跳信号时间，用于检测 GOOSE 传送是否中断，之前接收的信号状态是否还能用于逻辑判断。

- “启动信号” =1 事件发生后，装置会延时 2ms 从以太网接口发出。
- 再延时 T1=1ms 后再发送一次该“启动信号”。
- 再延时 T2=2ms 后再发送一次该“启动信号”。
- 再延时 T3=4ms 后再发送一次该“启动信号”。
- 之后就每隔 T0=500ms 重复发送“启动信号”。

C. 使用 GOOSE 原理实现差动保护功能存在误动的可能性

当 A 装置内的 B “启动信号” =0，这时 A 装置自己的“启动信号” =1，A 装置就判断有故障跳闸。保护就有可能误动，原因是：

- A 装置中的 B “启动信号” =0 的状态可能是 T0 周期时间之前的状态，由于以太网的延时、网络重构、网络瞬时堵塞、中断等原因，B 装置中的“启动信号” =1 还没有被送过来。
- B 装置的 GOOSE 心跳间隔时间为 T0=500ms，因此在 A 装置中用 1.5 倍的 T0 时间来检测 GOOSE 传送是否中断，因此 A 装置中的 B “启动信号” 会在延时 750ms 后才能被检测出来有问题，闭所保护跳闸。
- 因此从原理上分析，在 1.5*T0 时间间隔内判断为区内故障就有误动的可能性，这在原理上无法避免。
- 为了避免误动，差动出口时间就要设定延时（大于 1.5*T0），但这会造成差动出口慢的问题；如果减小 T0 时间会造成网络流量增大，CPU 负荷提高，网络的延时会更加不确定。

D. 问题的解决

让 A 装置差动出口信号平时一直处于闭锁状态，出口的解锁信号由对侧 B 装置产生，用 GOOSE 送到本侧 A 装置中做跳闸解锁。这样能让 GOOSE 的 T0 不影响差动的出口速度，T0 时间可以放在 2s 以上，还能减少装置 CPU 和网络负荷。

信号定义如下：

APU: A 侧过流保护启动 pickup (理论上故障发生后 20ms 产生此信号)。

BPU: B 侧过流保护启动 pickup (理论上故障发生后 20ms 产生此信号)。

APU-B: B 侧通过 GOOSE 接收到的 A 侧过流保护启动信号 (理论上会延期约 2ms 在 B 装置中生成逻辑 1)。

AOL-B: B 侧装置看见 A 设备在线。

BOL-A: A 侧装置看见 B 设备在线。

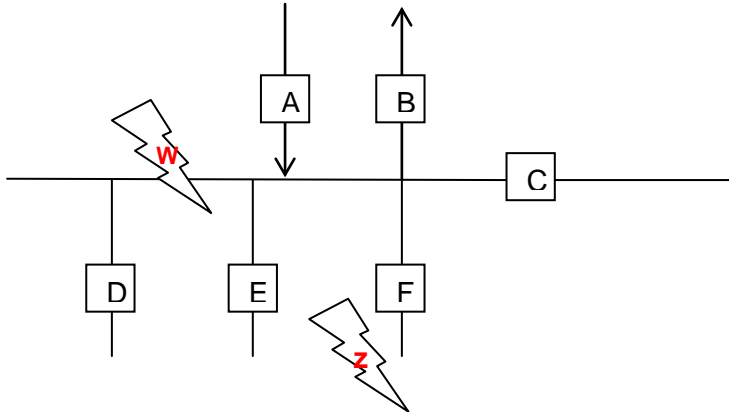
BUBLK: 在 B 装置内生成的开放差动出口的 unblock 信号, 逻辑: $BUBLK = AOL-B \& APU-B \& (! BPU)$, 并延时 10ms 生成信号。

BUBLK-A: A 装置收到的 B 装置发来的开放差动出口信号。

案例 1: 线路保护, 故障点 W

1. A 侧过流保护启动, $APU=1$, 通过 GOOSE 发送给 B 装置。
2. B 侧过流保护未启动 $BPU=0$, 收到 APU 信号, $APU-B=1$
3. B 侧装置 $BUBLK = [AOL-B \& APU-B \& (! BPU)]$ 延时 10ms = $1 \& 1 \& (! 0) = 1$, 并发送到 A 装置。延时 10ms 的目的是防止某种情况 B 装置过流启动慢, 如正好到定值, 或两侧过流返回时间的差异。
4. A 装置收到的 B 装置发来的开放差动出口信号 $BUBLK-A=1$, 差动出口逻辑 = $BOL-A \& APU \& BUBLK-A = 1 \& 1 \& 1 = 1$ 出口, 同时用 GOOSE 向 B 断路器发跳令。
5. 出口时间: $20ms \text{ APU} + 2ms \text{ B 装置收到产生 APU-B} + 10ms \text{ BUBLK 生成时间} + 2ms \text{ A 装置收到 BUBLK-A} + \text{A 继电器出口时间 } 6ms = 40ms$, 故障切除时间 $40ms + 60ms \text{ 断路器出口时间} = 100ms$, 上游后备过流级差配合 200ms。
6. 关键点是: 处于电源侧的 A 装置出口信号是由 A 装置自动触发, 且网络及 B 装置正常才会收到 unblock 信号, 这意味跳令是所有环节正常时产生的, 安全可靠。

案例 2：母线保护



A,B 为进出线,C 为母联, 它们都被看成电源端。D,E,F 为出线。所有断路器合位。

案例：故障 W

1. A、B、C 将自己生产的过流保护启动信号，发给除自己以外的所有装置。
2. 每个装置中对收到的每个启动信号都有一个对应的 unblock 信号。
3. 案例中 APU=1，通过 GOOSE 发送给 B、C、D、E、F 装置。
4. B、C、D、E、F 侧过流保护未启动，收到 APU 信号
5. B、C、D、E、F 侧装置都生成 unblock 并发送给 A 装置。
6. A 装置收到的 B、C、D、E、F 装置发来的开放差动出口信号，差动出口逻辑= $APU \& (BOL-A \& BUBLK-A) \& (COL-A \& CUBLK-A) \& (DOL-A \& DUBLK-A) \& (EOL-A \& EUBLK-A) \& (FOL-A \& FUBLK-A) = 1$ 出口跳 A 断路器。同时用 GOOSE 向 B、C、D、E、F 断路器发跳令。

案例：故障 Z

1. A 装置收到的 F 装置发来的开放差动出口信号 $FUBLK-A = 0$ ，差动出口逻辑= $APU \& (BOL-A \& BUBK-A) \& (COL-A \& CUBK-A) \& (DOL-A \& DUBK-A) \& (EOL-A \& EUBK-A) \& (FOL-A \& FUBK-A) = 0$ ，不出口跳 A 断路器。

按上述方法实现的电流闭锁式差动保护可以将 GOOSE 时间参数设为：

$T1=1ms$, $T0=2s$ 。