Strategie di gestione degli errori per un framework di applicazioni MOST

Alexander Leonhardi
Andreas Vallentin
Torsten Pech

Daimler AG

In questo articolo saranno discusse le cause degli errori in un sistema MOST e sarà descritta una strategia per rilevarli e gestirli sui diversi layer di un dispositivo MOST, incluso il layer delle applicazioni

Attualmente, i sistemi di infotainment automobilistici di fascia alta che assolvono alle funzioni di navigazione, intrattenimento e comunicazione in un veicolo sono di tipo distribuito, in cui i comandi di controllo e i contenuti audio/video e informativi vengono trasmessi tra diversi dispositivi specializzati sfruttando un bus MOST [2]. Ad esempio, un sistema comprende una cosiddetta Head-Unit (unità principale), che funge da interfaccia utente per il conducente, e diversi dispositivi periferici, ad esempio un sintonizzatore satellitare e un gateway al riproduttore multimediale dell'utente, controllati dall'Head-Unit.
Solitamente i sistemi di infotainment automobilistici non sono critici per quanto riguarda la sicurezza. Tuttavia, l'interattività delle applicazioni è elevata, ad esempio durante la navigazione nei titoli musicali di un riproduttore multimediale mobile. Errori come le perdite di messaggi possono avere diverse conseguenze sgradite per l'utente, ad esempio l'esecuzione ritardata di una funzione, la visualizzazione di informazioni errate, la mancata disponibilità di alcune funzioni, fino al blocco dell'applicazione. Di conseguenza la gestione degli errori nei sistemi di infotainment automobilistici mira principalmente a garantire che la relativa funzionalità sia offerta all'utente con un livello di reattività adeguato. Un'efficace gestione degli errori contribuisce significativamente alla qualità percepita del sistema. Inoltre, la gestione degli errori diventa importante anche in termini di sicurezza quando si tratta di gestire segnali audio, poiché un forte rumore improvviso dovuto a un guasto della rete di comunicazione può distrarre il conducente. In futuro la gestione degli errori diventerà sempre più importante, alla luce della tendenza attuale a integrare servizi di assistenza con i sistemi di infotainment.
La gestione degli errori è stata trattata ampiamente nei sistemi distribuiti, ad esempio per i sistemi critici per la sicurezza, per garantire la coerenza dei dati nei sistemi di database e per gli algoritmi distribuiti [6]. Tuttavia questi mecca-

nismi richiedono solitamente supporto hardware o aumentano significativamente i requisiti di comunicazione ed elaborazione. Date le caratteristiche sia del bus MOST, con il suo specifico layer fisico e di connessione dati, sia quelle specifiche del conducente - i NetServices [4], e in considerazione del fatto che lo standard MOST specifica il layer di applicazioni con una sintassi e una semantica specifiche (es. per la gestione delle notifiche), una strategia personalizzata di gestione degli errori estesa all'intero sistema è un elemento fondamentale di un framework di applicazioni MOST.

Nella prima parte di questo articolo saranno discussi i diversi tipi di errori che possono verificarsi in un sistema MOST, proponendone una classificazione in base alle loro cause e probabilità. Sulla base di tale classificazione, nella seconda parte verrà descritta una strategia di individuazione e gestione degli errori sperimentata che definisca a) i meccanismi per rimediare a un errore e b) i meccanismi per la gestione degli errori a cui non è stato possibile rimediare. La discussione sarà limitata a MOST25 e al suo layer fisico ottico. Inoltre, per motivi di brevità non verranno trattate le strategie di gestione degli errori in caso di errori multipli, di gestione delle connessioni e per il MOST High Protocol [1].

Cause degli errori in un sistema MOST

In un sistema MOST gli errori possono verificarsi per diverse ragioni e dovranno essere quindi identificati e gestiti sui diversi layer di un dispositivo MOST. Per discutere la gestione degli errori, verrà utilizzato il modello semplificato a 3 layer di un dispositivo MOST rappresentato in figura 1.

Di seguito sono elencate le ragioni più comuni di errori in un sistema MOST.

- Perdite di messaggi: sono causate principalmente da un dispositivo col buffer di ricezione pieno. Data la presenza di un solo buffer di ricezione in una scheda NIC (Network Interface Controller) [6], le perdite di singoli messaggi sono comuni. Anche le perdite di più messaggi non sono insolite quando un dispositivo si trova ad affrontare un carico elevato di elaborazione per le comunicazioni o applicazioni.
- Errori di trasmissione: gli errori di bit sulla rete fisica possono causare messaggi corrotti a un layer superiore. Tuttavia, dato il layer fisico ottico del bus MOST, gli errori di bit sono molto rari, tranne nel caso di situazioni instabili, ad esempio durante l'avvio.
- Comunicazioni interrotte: uno sblocco sul bus MOST determina un'interruzione delle comunicazioni. Gli sblocchi sono assai comuni, specialmente durante l'avvio del veicolo. Sui layer superiori, uno sblocco viene percepito come un burst di perdite di messaggi.
- Errori di configurazione: i dispositivi che si uniscono o abbandonano un sistema MOST, aprendo o chiudendo il loro bypass determinano una modifica nella configurazione del sistema, detto evento MPR. Le modifiche di configurazione possono verificarsi assai spesso (ciò succederà più raramente con l'attuale scheda NIC [5] rispetto a quella precedente), ad esempio durante l'avvio del veicolo.
- Indisponibilità delle funzioni dei dispositivi: a causa di un carico di elaborazione elevato, di un problema su un sistema esterno o di un errore specifico su un dispositivo, la funzione di un dispositivo necessario per eseguire una richiesta dell'utente potrebbe non essere attualmente disponibile. La probabilità che si verifichi un errore di questo tipo dipende in gran parte dal tipo di dispositivo. Tali errori devono essere gestiti sul layer delle applicazioni.
- Guasto di dispositivi ed errori di implementazione, ad esempio quando un dispositivo invia il messaggio di errore sbagliato: tale problema si presenta spesso durante l'integrazione e il collaudo del sistema e verrà eliminato nel sistema pronto alla produzione.

Tuttavia, alcuni errori possono presentarsi anche nei sistemi pronti, ad esempio a causa di limiti di temporizzazione non idonei (es. in condizioni di competizione). Purtroppo tali errori possono essere identificati solo tramite i loro "sintomi", di conseguenza la causa dell'errore è determinabile solo come ipotesi plausibile. Alcuni errori potrebbero non venire affatto identificati.

Individuazione e gestione degli errori

La specifica MOST [2] definisce già alcuni requisiti per la gestione degli errori in un sistema MOST. Ad esempio, la Sezione 2.2.3.5.1 definisce un insieme dettagliato di risposte di errore per i messaggi di controllo. Tuttavia, tali requisiti descrivono solo i meccanismi necessari a garantire l'interoperabilità di un sistema MOST. Tocca all'integratore del sistema definire una strategia coerente di gestione degli errori.

Panoramica

La figura 2 riporta il diagramma operativo di gestione degli errori. Sostanzialmente, tale diagramma viene eseguito su ciascun layer a partire da quello fisico. Gli errori verranno rilevati sul layer appropriato (es. certi errori possono essere identificati solo sul layer delle applicazioni) e da lì inizierà la gestione dell'errore. Sarà necessario implementare diverse strategie di risoluzione per le classi di errore definite sopra, per considerare la probabile causa dell'errore. Gli errori che non possono essere gestiti su un layer vengono passati al layer superiore successivo.

Layer fisico

Gli errori di base che si verificano sul layer di connessione fisica e di collegamento dati del bus MOST vengono gestiti dal layer fisico. Le schede NIC MOST garantiscono già un meccanismo in grado di rilevare le perdite di

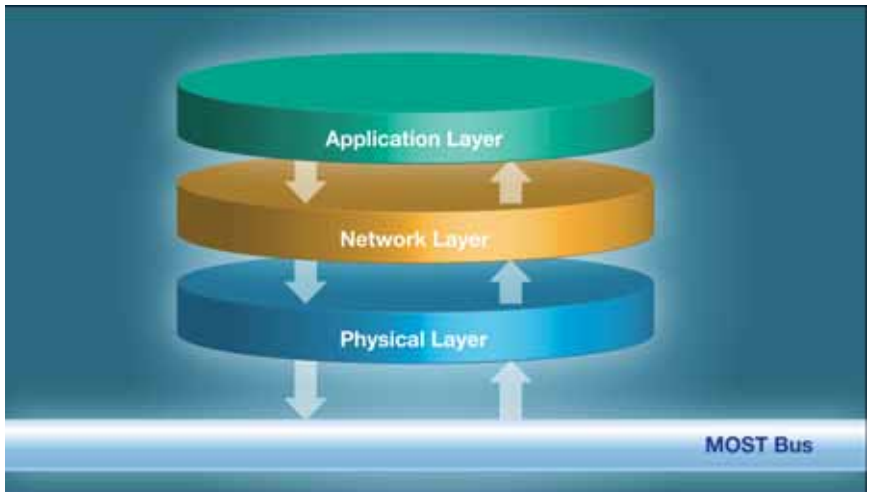

Fig. 1 - Modello semplificato a 3 layer di un dispositivo MOST

messaggi tramite un meccanismo ACK/NAK sul canale di controllo MOST. Se un ricevitore non riconosce un messaggio, il mittente innesca un numero specificato di tentativi ripetuti. Nonostante si possano evitare i tentativi ripetuti a livello superiore specificando un gran numero di tentativi ripetuti a livello inferiore, ciò determina un carico elevato sul bus e blocca il mittente per la durata dell'operazione, aumentando la probabilità di uno stallo.
Gli errori di trasmissione vengono identificati tramite un meccanismo CRC (Cyclic Redundancy Check), determinando anche in questo caso tentativi ripetuti a basso livello. Dato il verificarsi assai raro di errori di trasmissione, solitamente non è necessario gestirli a un layer superiore. Inoltre, il layer fisico consente di gestire una comunicazione interrotta a causa di uno sblocco. Dopo 70 ms o una serie di brevi sblocchi, il layer fisico segnala uno sblocco critico che conduce a una chiusura per errore del bus MOST.
Di conseguenza la gestione dell'errore deve essere in grado di affrontare burst di perdite di messaggi fino a 70 ms.

Layer di rete

Il layer di rete garantisce l'ulteriore gestione degli errori tramite i NetServices e gestisce le modifiche alla rete.
Il Network Master è responsabile del rilevamento e della gestione delle modifiche alla configurazione. Quando rileva una modifica alla rete determinata da un evento MPR, il Network Master innesca una scansione di rete e notifica le modifiche a tutte le applicazioni nel sistema MOST. La stabilità del Network Master è fondamentale per la stabilità del sistema, specialmente se ci sono diverse modifiche alla configurazione durante l'avvio.
I possibili tentativi ripetuti a livello medio innescati dai NetServices possono essere utilizzati per espandere l'intervallo temporale coperto dai tentativi ripetuti a basso livello. Essi consentono di utilizzare un minor numero di tentativi ripetuti a basso livello, aumentando così le prestazioni del sistema. Insieme, i tentativi ripetuti a medio e basso livello dovrebbero essere in grado di affrontare le perdite di messaggi che si verificano per i 70 ms prima che sia causato uno sblocco critico.

Layer delle applicazioni

Gli errori a un livello superiore dei protocolli MOST, come ad esempio una mancata risposta a una richiesta con OPType Get, possono essere gestiti solo dal layer delle applicazioni, poiché non vengono rilevati ai layer inferiori. Inoltre, i tentativi ripetuti a livello delle applicazioni possono risolvere situazioni che condurrebbero altrimenti a uno stallo, poiché consentono l'esecuzione delle altre richieste attualmente presenti nel buffer di invio di NetServices prima che la richiesta venga nuovamente eseguita. Di conseguenza, i dispositivi MOST devono implementare i meccanismi di gestione degli errori anche sul layer delle applicazioni. Per un'ulteriore discussione della gestione degli errori a livello delle applicazioni, si veda anche la discussione riguardante l'argomento end-to-end in [3].
In primo luogo, il layer delle applicazioni deve implementare un meccanismo di timeout con rispettiva gestione degli errori per il meccanismo di risposta alle richieste definito dal protocollo di controllo MOST.
- Per richieste a proprietà con OPType Get/SetGet nonché per richieste a metodi con OPType StartResult/StartResultAck, la specifica MOST indica un meccanismo di timeout appropriato. Se il timeout scade, viene applicata la gestione degli errori appropriata. Come strategia di recupero, il layer delle applicazioni deve eseguire un tentativo ripetuto. Se fallisce, la richiesta corrente verrà abbandonata e l'applicazione tornerà al funzionamento normale supponendo una risposta valida.
Un'ulteriore responsabilità della gestione degli errori sul layer delle applicazioni è una risposta di errore ricevuta per una richiesta. Le risposte di errore per il protocollo di controllo sono definite nella sezione 2.2.3.5.1 della specifica MOST [2]. Gli elementi elencati di seguito offrono una panoramica della gestione degli errori per le risposte di errore più importanti.
- Gli errori "FBlockID not available" (FBlockID non disponibile, codice errore: 0x01) e "InstId not available" (InstId non disponibile, 0x02) indicano un errore di configurazione.

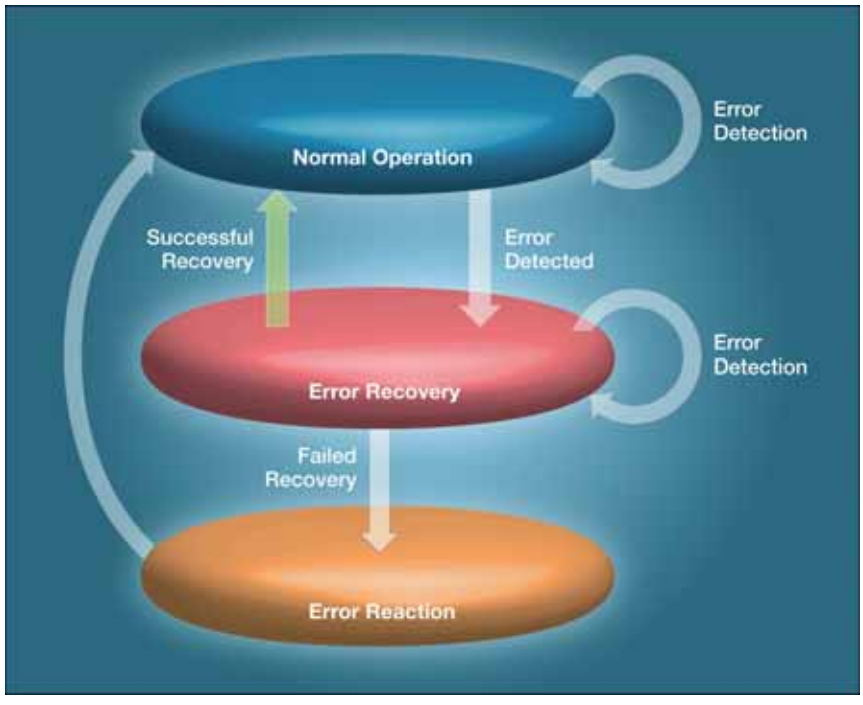

Fig. 2 -Diagramma di stato per i meccanismi di gestione dell'errore

Di conseguenza, se un primo tentativo ripetuto a livello delle applicazioni è fallito, il dispositivo deve aggiornare le sue informazioni di registro ed eseguire, se è ancora possibile, un altro tentativo. Se il secondo tentativo ripetuto è fallito, la mancata disponibilità della funzionalità deve essere segnalata all'utente.

- Gli ulteriori errori "FktID not available" (FktID non disponibile, 0x03) e "OPType not available" (OPType non disponibile) e gli errori di parametro (da 0x05 a 0x07), indicano un problema interno di un dispositivo o un problema di implementazione. In questo caso, il layer delle applicazioni deve eseguire un tentativo ripetuto.
- La gestione degli errori "Function specific" (specifici di funzione, 0x20) deve essere definita nel catalogo delle funzioni della rispettiva funzione. Se non è specificato nulla, per impostazione predefinita deve essere attuato lo stesso comportamento adottato per l'ultimo elemento.
- La temporanea indisponibilità di una funzione di un dispositivo può essere indicata dagli errori "Busy" (Occupato, 0x40), "Not available" (Non disponibile, 0x41) o "Processing errore (Errore di elaborazione, 0x42). In questo caso, sarà necessario eseguire un maggior numero di tentativi ripetuti con un intervallo temporale più ampio per risolvere l'errore.

In questo articolo sono state discusse le cause degli errori in un sistema MOST e descritta una strategia per rilevarli e gestirli sui diversi layer di un dispositivo MOST, incluso il layer delle applicazioni.

Nonostante ci possano essere diverse strategie valide di gestione degli errori, quella descritta è stata implementata con successo nell'architettura del sistema MOST, dimostrandosi valida nei progetti di serie esistenti.

Uno degli obiettivi del presente articolo è aumentare la consapevolezza dell'importanza della gestione degli errori in un sistema MOST e avviare una discussione più approfondita riguardante i meccanismi specifici di gestione degli errori. Poiché è bene che la gestione degli errori in un sistema MOST si attenga a una strategia uniforme, si ritiene che il supporto per la gestione degli errori debba essere incluso nel software di serie di un framework di applicazioni MOST.

Se in futuro il bus MOST sarà utilizzato anche per i sistemi di assistenza al conducente, ad esempio per trasmettere le immagini video di un sistema di telecamere, potrebbe essere necessario definire strategie di gestione degli errori più elaborate per i casi di utilizzo che richiedano un livello maggiore di tolleranza ai guasti.

MOST
www.mostcooperation.com

Bibliografia

[1] MOST Cooperation: MOST High Protocol Specification, Rev. 2.2, URL: www.mostcooperation. com, Aug. 2005
[2] MOST Cooperation: MOST Specification, Rev. 3.0, URL: www.mostcooperation.com, Jun. 2008
[3] J. Saltzer, D. Reed, and D.D. Clark: End-to-End Arguments in System Design, ACM Transactions on Computer Systems, Vol. 2, No. 4, Nov. 1984, pp. 277-288
[4] SMSC Cooperation: MOST NetServices Layer I, Wrapper for INIC; V2.1.x, User Manual Specification, URL: www.smsc-ais.com, Jul. 2008
[5] SMSC Cooperation: OS81050 Data Sheet, URL: www.smsc-ais.com, Aug. 2007
[6] G. Tel: Introduction to Distributed Algorithms, Second Edition, Cambridge University Press, Feb. 2000