

Manuale Operativo del Monitor di Rete (Network Monitor)

Il presente manuale descrive il funzionamento e la configurazione del sistema di monitoraggio. Il sistema è progettato per monitorare periodicamente la latenza di host specifici e fornire un feedback visivo e acustico sullo stato della rete.

1. Panoramica e Funzionamento Generale

Il sistema, implementato nella classe `Form1`, utilizza le funzionalità di rete (`System.Net.NetworkInformation`) per eseguire ping periodici sugli host configurati.

- 1. Avvio e Configurazione:** All'avvio del form, il sistema esegue la routine `LoadConfig` per caricare le impostazioni.
- 2. Temporizzazione:** Il monitoraggio viene eseguito a intervalli definiti da `Timer1`. L'intervallo predefinito è impostato a 5000 millisecondi (5 secondi), ma è configurabile.
- 3. Monitoraggio (Tick):** Ad ogni intervallo del timer, il sistema invia un comando `Ping.Send` a tutti gli host configurati.

2. Configurazione degli Host

La configurazione degli host monitorati avviene tramite la lettura di un file di testo denominato `config.txt`.

2.1 Struttura del File di Configurazione

Il sistema legge le righe del file ignorando quelle vuote o che iniziano con il simbolo cancelletto (`#`). Ogni riga di configurazione host valida deve essere suddivisa in campi tramite il punto e virgola (`;`) e deve contenere almeno quattro parti.
Le configurazioni definiscono le proprietà della classe `HostConfig1`:

Posizione	Proprietà	Descrizione	Esempio
1	Host	Indirizzo IP o nome DNS dell'host da monitorare.	<code>google.com</code>
2	GoodThreshold	Latenza massima (ms) per considerare lo stato " Buono ".	<code>80</code>
3	WarnThreshold	Latenza massima (ms) per considerare lo stato di " Attenzione ".	(intero)
4	SoundEnabled	Determina se gli avvisi sonori sono abilitati per questo host specifico (" <code>yes</code> " per abilitare).	<code>yes</code> o <code>no</code> (o <code>True</code> / <code>False</code> se si basa sull'esempio 14, anche se il codice converte esplicitamente " <code>yes</code> " a <code>True</code>)

2.2 Esempio di Record di Configurazione

Un host configurato come `google.com` con soglie di latenza di 80 ms (Good) è presente nelle configurazioni di esempio.

3. Interfaccia Utente e Stati Visivi

Per ogni host configurato, il sistema crea un pannello dedicato. Questo pannello visualizza lo stato operativo e la latenza attraverso un sistema di codifica a colori e grafici.

3.1 Componenti del Pannello Host

Ogni pannello creato (`CreateHostPanel`) include:

- Un'etichetta (`Label`) con il nome dell'host.
- Un indicatore visivo (`PictureBox`) che riflette lo stato.
- Un pulsante "📊" (`toggleChartBtn`) per alternare la visibilità del grafico associato.
- Un Grafico (`Chart`) per la visualizzazione storica della latenza.

3.2 Codifica Colore (Funzione `UpdateUI`)

Lo sfondo del pannello cambia colore in base alla latenza misurata:

Colore	Condizione	Azione Correlata	Messaggio di Log (Esempi)
Verde	<code>Latenza < conf.GoodThreshold</code>	Stato Ottimale	X ms
Giallo	<code>Latenza < conf.WarnThreshold</code> ma maggiore o uguale a <code>GoodThreshold</code>	Stato di Avviso (Warning)	X ms (warning)
Arancione	<code>Latenza > conf.WarnThreshold</code>	Stato di Latenza Alta	X ms (high latency)
Rosso	<code>Latenza < 0</code> (Timeout o Fallimento Ping)	Stato di Errore	Timeout

In caso di stato **Arancione** o **Rosso**, viene riprodotto un suono di avviso se abilitato.

3.3 Gestione del Grafico

Il grafico, inizialmente nascosto (`Visible = False`), visualizza la serie di dati "Ping" come un grafico a linee (`SeriesChartType.Line`).

- **Dati:** I dati storici (`ChartData`) sono gestiti come una coda (`Queue(Of Long)`). Se la coda raggiunge 10 valori, il valore più vecchio viene rimosso prima di aggiungerne uno nuovo, mantenendo un registro degli ultimi 10 risultati di ping.
- **Visibilità:** Cliccando sul pulsante "📊", il grafico viene reso visibile o nascosto, e l'altezza del pannello viene adattata (140 quando visibile, 60 quando nascosto).

3.4 Interazione Utente

Clic sul Pannello: Se l'utente clicca su un pannello host, viene avviato un prompt dei comandi (`cmd.exe`) eseguendo un ping continuo (`/k ping <host>`) verso quell'host.

4. Funzionalità Aggiuntive

4.1 Log

Il sistema esegue una registrazione degli eventi (Log) per ogni cambiamento di stato o risultato del ping.

- **Posizione:** I file di log vengono salvati in una sottodirectory denominata "**Logs**", che viene creata nella directory base dell'applicazione se non esistente.
- **Nome File:** Il nome del file di log è strutturato come `log_AAAA MM GG_hostsanitizzato.txt`. L'host viene "sanitizzato" (punti, due punti, backslash e slash sono sostituiti con underscore) per garantire la validità del nome file.
- **Contenuto:** Ogni riga di log registra l'orario (`HH:mm:ss`) e un messaggio descrittivo dello stato (es. "Timeout" o la latenza in ms).

4.2 Controllo Audio

Il sistema può riprodurre il suono di sistema "Exclamation" (esclamazione) in caso di timeout o latenza alta.

- **Condizioni di Riproduzione:** Un suono viene riprodotto solo se **sia** la proprietà `SoundEnabled` dell'host configurato **sia** la variabile globale `SoundEnabledGlobal` sono impostate su `True`.

- **Controllo Globale:** La variabile `SoundEnabledGlobal` è gestita da un controllo chiamato `chkMuteAll`. Lo stato globale è `True` se `chkMuteAll` **non è spuntato** (non è muto).

Il file di configurazione (`config.txt`) contiene una sezione dedicata all'integrazione con Telegram. Questa sezione si trova nel file di configurazione dopo le definizioni degli host e serve per abilitare e configurare l'invio di notifiche tramite l'API di Telegram.

Configurazione di Telegram

La sezione Telegram nel file `config.txt` è definita dai seguenti parametri:

Parametro	Valore di Esempio	Descrizione e Utilizzo
[Telegram]		Intestazione della sezione di configurazione.
Enabled	<code>True</code>	Indica se l'invio di notifiche Telegram è attivo o disattivo.
BotToken	<code>808731xxxx:xxxxDnCwRtUqUgUy xxxxtQWfTQZlIw6UYiY</code>	La chiave (token) univoca fornita da BotFather al momento della creazione del bot. È essenziale per autenticare le richieste API del bot.
ChatID	<code>1xxxxxxxx21</code>	L'identificatore numerico della chat (o del gruppo) in cui il bot deve inviare i messaggi di notifica.

Utilizzo del Bot per le Notifiche

`BotToken` e `ChatID` il sistema è progettato per inviare avvisi agli utenti Telegram.

L'uso tipico di questa configurazione avviene in corrispondenza degli eventi di monitoraggio critici:

1. **Eventi Critici:** Quando un host entra in uno stato di fallimento (Timeout, latenza <0) o di latenza alta (latenza superiore a `WarnThreshold`).

2. **Azione:** Invece di limitarsi a cambiare il colore del pannello in arancione o rosso e riprodurre un suono di sistema, l'applicazione utilizza il `BotToken` e il `ChatID` per inviare un messaggio testuale al canale o all'utente specificato.

In sintesi, affinché il sistema utilizzi Telegram, è necessario che il parametro `Enabled` sia impostato su `True` e che l'utente abbia inserito il proprio `BotToken` e il `ChatID` corretto, permettendo al bot di notificare immediatamente gli stati di "Timeout" o "High Latency" registrati nei log.