บทที่ 2

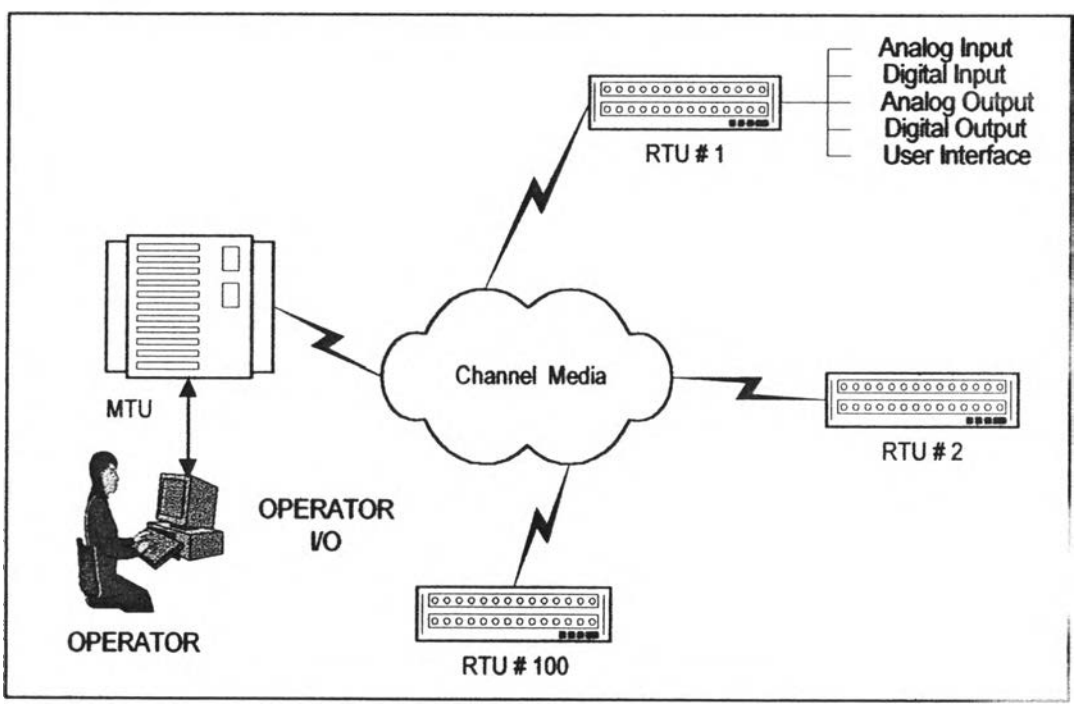
ระบบสเกดา

2.1 ระบบสเกดา (SCADA System)

ระบบสเกดาเป็นเทคโนโลยีที่ทำให้ผู้ใช้สามารถเก็บรวบรวมข้อมูลจากสถานที่แห่งเดียวหรือหลายๆแห่งเป็นระยะทางไกลๆ ออกไปและสามารถส่งคำสั่งควบคุมจากศูนย์กลาง โดยที่ระบบสเกดาทำให้ผู้ปฏิบัติงานสามารถปฏิบัติงานได้ที่สถานีศูนย์กลางในสภาวะการทำงานปกติไม่จำเป็นที่จะต้องประจำอยู่ที่สถานที่ไกลๆ

คำจำกัดความของสเกดา (SCADA) [4][8][9]

คำว่า SCADA ย่อมาจาก SUPERVISORY CONTROL AND DATA ACQUISITION ซึ่งเป็นคำนิยามที่แปลความหมายได้ตรงตัว คือเก็บรวมรวมข้อมูล และควบคุม แต่ในที่นี้ไม่ได้มีการกล่าวถึงปัจจัยของระยะทางซึ่งเป็นที่เข้าใจกันอยู่แล้วสำหรับระบบสเกดาว่าเป็นระบบที่ใช้สำหรับในกรณีของระยะทางเป็นปัจจัยสำคัญ เช่น ระบบสายส่งไฟฟ้าแรงสูง, ระบบท่อส่งน้ำมันหรือก๊าซ เป็นต้น ข้อมูลที่เก็บคือข้อมูลจากการวัดค่าต่างๆ หรือสัญญาณเตือนภัยต่างๆ จากสถานีปลายทาง ในส่วนของการควบคุมอาจใช้ควบคุมการเปิด-ปิดวาล์ว หรือสวิตช์



รูปที่ 2.1 ระบบสเกดา

รูปที่ 2.1 แสดงให้เห็นถึงองค์ประกอบหลักของระบบสเกดา ซึ่งสถานีศูนย์กลางคือ จุดที่ผู้ปฏิบัติงาน (Operator) จะติดต่อสื่อสารกับระบบผ่านทางเครื่องไมโครคอมพิวเตอร์ (Operator I/O)

ตัวดำเนินการส่งข้อมูลออก หมายถึงจอมอนิเตอร์ซึ่งทำหน้าที่แสดงผล

ตัวดำเนินการรับข้อมูลเข้า หมายถึงคีย์บอร์ดจากนั้นจะทำหน้าที่รับคำสั่งหรือรับข้อมูลจากผู้ปฏิบัติงานส่งไปยังระบบ

MTU (Master Terminal Unit) ซึ่งหมายถึงส่วนควบคุมระบบ โดยทั่วไปจะเป็นเครื่องไมโครคอมพิวเตอร์

MTU ทำการติดต่อสื่อสารกับ RTU (Remote Terminal Unit) ที่ติดตั้งอยู่ไกลออกไปจากสถานีศูนย์กลาง การสื่อสารระหว่าง MTU กับ RTU จะติดต่อสื่อสารผ่านช่องสื่อสาร (Channel Media) การสื่อสารระหว่าง MTU กับ RTU ขึ้นอยู่กับผู้ใช้งานในการที่จะเลือกใช้ช่องทางการสื่อสารใดๆ เช่น รับ-ส่งทางคลื่นวิทยุ, ผ่านโมเด็มสายโทรศัพท์, ผ่านสายสัญญาณ, ผ่านไมโครเวฟ เป็นต้น

ระบบสเกดาเป็นระบบที่สำหรับใช้งานในการควบคุมและตรวจสอบข้อมูลจาก MTU ผ่านทาง RTU เช่น ระบบท่อส่งน้ำมันในกรณีนี้เราจำเป็นต้องมีระบบสเกดาในการควบคุมและตรวจสอบข้อมูลจากจุดตรวจสอบของท่อส่งหลายๆ จุดเนื่องจากว่าเราไม่สามารถที่จะส่งเจ้าหน้าที่ไปประจำในแต่ละจุดทุกๆ 10 กิโลเมตรเพื่อคอยตรวจสอบข้อมูลการไหลเข้าและการไหลออกแต่ละจุดได้ การที่จะตรวจสอบว่ามีน้ำมันรั่วออกจากท่อหรือไม่สามารถทำได้โดยง่ายคือ เรามีเครื่องมือวัดอัตราการไหลของน้ำมันเข้าและวัดอัตราการไหลออกถ้าอัตราการไหลเข้าเท่ากับอัตราการไหลออก นั่นแสดงว่าไม่มีการรั่วไหลของน้ำมันในท่อ แต่ถ้าอัตราการไหลเข้าไม่เท่ากับอัตราการไหลออกแสดงว่ามีน้ำมันรั่วเกิดขึ้นที่ช่วงของการวัด เมื่ออุปกรณ์ RTU หมายเลข 1 จะทำหน้าที่ในการส่งข้อมูลที่ได้จากการวัดค่าของอัตราการไหลเข้าและออกของน้ำมัน ณ จุดนั้น ส่งกลับไปยังMTU และอุปกรณ์ RTU หมายเลข 2 ก็ส่งข้อมูลแสดงอัตราการไหลเข้าและออกของน้ำมัน ณ จุดที่อยู่ห่างออกไป 10 กิโลเมตร จากนั้น MTU จะนำข้อมูลที่ได้ไปประมวลผลและแสดงข้อมูลให้ทางผู้ควบคุมที่ศูนย์กลาง ซึ่งผู้ควบคุมอาจพบสิ่งผิดปกติคือ อัตราการไหลออกของ RTU หมายเลข 1 มีค่ามากกว่าอัตราการไหลเข้าของ RTU หมายเลข 2 นั่นแสดงให้เห็นว่าจะต้องมีการรั่วไหลของน้ำมันออกจากท่อในระยะทางระหว่าง อุปกรณ์ RTU หมายเลข 1 และอุปกรณ์ RTU หมายเลข 2 ผู้ควบคุมสามารถที่จะส่งคำสั่งควบคุมจากศูนย์กลางมายัง RTU ที่เกิดปัญหานั้นเพื่อปิดวาล์วส่งน้ำมันที่ RTU หมายเลข 1 เพื่อป้องกันการเกิดปัญหาและในขณะเดียวกันก็สามารถส่งเจ้าหน้าที่ซ่อมบำรุงไปตรวจสอบและแก้ไข ณ จุดระหว่าง RTU หมายเลข 1 และ RTU หมายเลข 2 ได้ทันเวลา จากตัวอย่างแสดงให้เห็นว่าเราสามารถที่จะใช้ระบบสเกดาในการควบคุมและสั่งการได้จากศูนย์กลางโดยมีอุปกรณ์ RTU ติดตั้งกระจายอยู่หลายๆจุด ทำให้เราสามารถลดกำลังเจ้าหน้าที่ที่จะต้องไปประจำอยู่ตามจุดต่างๆ เพื่อทำงานเพียงแค่รับส่งข้อมูล และ เปิดหรือปิดวาล์ว เท่านั้น

ระบบสเกดาที่ใช้ภายในโรงงานอุตสาหกรรมส่วนใหญ่จะเป็นระบบที่ส่งข้อมูลจาก RTU เพื่อเก็บสถิติของข้อมูลการผลิต, ตรวจสอบปริมาณการผลิตและ ตรวจสอบการทำงานของเครื่องจักร เป็นต้น ช่วยในการบริหารและจัดการเกี่ยวกับการผลิตทางอุตสาหกรรม ทำให้การผลิตเป็นไปอย่างมีประสิทธิภาพ

ระบบสเกดาสามารถใช้ในระบบควบคุมพลังงาน (Energy Management System) โดย ข้อมูลการใช้พลังงานในแต่ละจุดที่ RTU ส่งข้อมูลกลับมายัง MTU มาทำการวิเคราะห์การใช้พลังงานในช่วงเวลาต่างๆ เราสามารถนำสิ่งที่วิเคราะห์มาบริหารการใช้พลังงานเพื่อเป็นการใช้พลังงานได้อย่างคุ้มค่าและลดค่าใช้จ่ายสำหรับที่ต้องจ่ายในกรณีมีภาระการใช้พลังงานสูงสุดในแต่ละวันได้

ระบบสเกดาเป็นระบบที่จะต้องลงทุนค่อนข้างสูงตอนเริ่มต้น แต่ก็สามารถให้ผลตอบแทนที่คุ้มค่าในระยะยาว ไม่ว่าจะเป็นการลดค่าใช้จ่ายด้านกำลังคน, ลดค่าใช้จ่ายในการบริหารโครงการ, เพิ่มประสิทธิภาพในการใช้พลังงานอย่างคุ้มค่า, เพิ่มประสิทธิภาพในการบริหารการผลิตทางอุตสาหกรรม เป็นต้น

ดังนั้นการจะพิจารณาที่จะเลือกใช้ระบบสเกดาจะต้องพิจารณาอย่างรอบคอบเพราะเป็นการลงทุนในระยะปานกลางถึงระยะยาว

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2 องค์ประกอบของระบบสเกดา

จากรูปที่แสดง องค์ประกอบหลักของระบบสเกดาจะประกอบด้วย

1. MTU ซึ่งอาจจะใช้คอมพิวเตอร์เป็นอุปกรณ์หลักในการเชื่อมการสื่อสารระหว่างผู้ปฏิบัติงานกับ RTU

MTU ประกอบด้วย

- คอมพิวเตอร์ เป็นสำหรับประมวลผลเพื่อแสดงข้อมูลที่ได้รับจาก RTU ออกจอมอนิเตอร์ ในขณะเดียวกันก็รับคำสั่งจากผู้ปฏิบัติงานผ่านทางคีย์บอร์ดส่งไปยัง RTU
- ซอฟต์แวร์หรืออุปกรณ์อื่นๆ เพื่อรับ-ส่งข้อมูลผ่านช่องทางการสื่อสารไปยัง RTU

ซึ่งต่อไปเราจะเรียกแทน MTUและส่วนประกอบอื่นๆ ว่าเป็น สถานีแม่ข่าย

2. RTU ซึ่งเป็นอุปกรณ์ปลายทางสำหรับรับคำสั่งจากสถานีแม่ข่าย พร้อมกับส่งข้อมูลจากการวัดกลับไปยังสถานีแม่ข่าย ผ่านทางช่องทางการสื่อสาร

RTU ประกอบด้วย

- ส่วนควบคุมและแปลคำสั่งของRTU เป็นส่วนหลักที่เก็บข้อมูล และ สั่งงานโดยรับคำสั่งมาจากMTU พร้อมทั้งส่งข้อมูลต่างๆ ที่เก็บส่งกลับไปยังMTU
- ซอฟต์แวร์ ซึ่งจะเป็นโปรแกรมในการทำงานหลักของ RTU เพื่อเก็บข้อมูล, สั่งให้อุปกรณ์ต่างๆที่ใช้งานร่วมอยู่ทำงานตามคำสั่งจากสถานีแม่ข่าย รวมทั้ง รับ-ส่งข้อมูล กลับไปยังสถานีแม่ข่าย

ซึ่งต่อไปเราจะเรียกแทน RTU และส่วนประกอบอื่นๆ ว่าเป็นสถานีลูกข่าย

3. ช่องสื่อสารเป็นส่วนของการสื่อสารสำหรับ สถานีแม่ข่าย กับ สถานีลูกข่าย ซึ่งอาจรวมถึงอุปกรณ์ที่ใช้ในการเชื่อมต่อเช่น เครื่องรับ-ส่งสัญญาณวิทยุ, อุปกรณ์โมเด็ม เป็นต้น

2.3 โปรโตคอล (Protocol) ที่ใช้ในระบบสเกดา

โปรโตคอลคือข้อตกลงที่กำหนดขึ้นให้การรับ-ส่งข้อมูลไบนารีมีรูปแบบและความหมายแต่ละบิตในการรับส่งข้อมูลเพื่อให้เป็นที่เข้าใจกันระหว่างสถานีส่งและสถานีรับ ดังนั้นสิ่งที่สำคัญที่สุดก็คือการที่สถานีส่งและสถานีรับจะสามารถติดต่อสื่อสารเข้าใจกันได้ต้องมีโปรโตคอลเดียวกัน

การส่งข้อมูล อาจเป็นการส่งแบบ จุด ไปยัง จุด (Point to Point) หรือ จุด ไป หลายจุด (Point to Multipoint) หรือการแพร่กระจายทุกจุด(Broadcast) โดยส่งผ่านช่องทางการสื่อสารช่องทางเดียว ในการส่งข้อมูลผู้ใช้อาจใช้สื่อในการส่งข้อมูลต่างๆ เช่น สายเกลียวคู่ (Twisted pairs cable) สายโคแอกเชียล (Coaxial cable), สายใยแก้ว (Fiber Optic cable) หรือผ่านดาวเทียม หรือสัญญาณวิทยุ ขึ้นอยู่กับผู้ใช้งาน

การส่งข้อมูลจะเป็นแบ่งออกเป็น 2 ลักษณะการส่ง

1. การส่งแบบอะซิงโครนัส (Asynchronous)
2. การส่งแบบซิงโครนัส (Synchronous)

การส่งข้อมูลที่เราจะพิจารณาสำหรับระบบสเกดา เพื่อใช้กับโปรโตคอล จะเป็นการส่งข้อมูลแบบซิงโครนัส

(Synchronous) โดยจะแปลงจากการส่งผ่านพอร์ตสื่อสารแบบอนุกรม แล้วนำมาเก็บไว้ในลักษณะที่เรียกว่า เฟรมข้อมูล (Data frame) องค์ประกอบภายในของแต่ละเฟรมข้อมูลจะเหมือนกัน ทำให้เราสามารถตรวจสอบและแบ่งแยก ได้ง่ายว่า มีข้อมูลอะไรบ้างภายในเฟรมนั้นๆ

โปรโตคอลที่ใช้ในระบบสเกดา อาจจะมีผู้ผลิตหลายรายที่ใช้โปรโตคอลเฉพาะของตัวเอง (Proprietary protocol) ขึ้นมาเพื่อใช้ในระบบเฉพาะตัว เช่น MOTOROLLA ใช้โปรโตคอล MOSCAD, BRISTOL BABCOCK ใช้โปรโตคอล BSAP และ MODICON ใช้โปรโตคอล MOD BUS เป็นต้น โปรโตคอลของผู้ผลิตแต่ละรายไม่เป็นที่เปิดเผย ทำให้ผู้ใช้งานไม่สามารถขยายระบบหรือพัฒนาได้ด้วยตนเอง และยากที่จะทำการเชื่อมต่ออุปกรณ์ต่างยี่ห้อเข้าด้วยกันได้

ดังนั้นจึงเป็นที่มาที่เราจะพิจารณาถึง โปรโตคอลที่สามารถอ้างเป็นมาตรฐานเป็นที่ยอมรับและง่ายต่อการเชื่อมต่อระบบของแต่ละยี่ห้อเข้าด้วยกันโดยใช้มาตรฐานโปรโตคอลเดียวกัน คือ IEEE std 999-1992 Protocol ซึ่งเป็นโปรโตคอลที่ IEEE (Institute of Electrical and Electronics Engineers) ได้พัฒนาขึ้นเพื่อใช้กับระบบสเกดาสำหรับการส่งสัญญาณดิจิตอลแบบอนุกรม (Serial Digital Transmission) โดยจำกัดความเร็วของอัตราการส่งข้อมูลที่น้อยกว่า 10,000 บิตต่อวินาที ไม่เหมาะสมที่จะใช้กับระบบเครือข่ายที่ใช้ความเร็วสูง โดยได้รับการอนุมัติจาก IEEE Standards Board เมื่อวันที่ 18 มิถุนายน ค.ศ. 1992 และได้รับการอนุมัติจาก ANSI (American National Standards Institute) เมื่อวันที่ 12 มกราคม ค.ศ. 1993

ชนิดของโปรโตคอลที่เราจะพิจารณาเพื่อเปรียบเทียบเป็นโปรโตคอลที่ใช้เป็นมาตรฐาน และเปิดเผย ในระบบสเกดาทั่วไปมีดังนี้

2.3.1 HDLC (High Level Data Link Control)

2.3.2 IEEE std 999-1992

เหตุผลที่เราทำการเปรียบเทียบ โปรโตคอลของ IEEE std 999-1992 กับ โปรโตคอล HDLC

1. HDLC เป็นโปรโตคอลมาตรฐานที่เป็นที่ยอมรับกันโดยทั่วไป
2. HDLC เป็นโปรโตคอลที่มีลักษณะฟังก์ชันการใช้งานสำหรับระบบสเกดา
3. HDLC มีข้อแตกต่างอย่างชัดเจนกับ โปรโตคอล IEEE std 999-1992 เช่น Control field bits การตรวจสอบข้อผิดพลาดของการส่งข้อมูล ซึ่งโปรโตคอล IEEE std 999-1992 ได้แก้ไขข้อบกพร่องในการใช้งานของ HDLC กับระบบสเกดา ทำให้มีความเหมาะสมมากกว่าในการใช้งานเพื่อเป็นมาตรฐาน สำหรับระบบสเกดา ดังจะได้กล่าวในรายละเอียดของแต่ละ โปรโตคอลต่อไป

2.3.1 HDLC (High Level Data Link Control) [1] [5]

เป็นโปรโตคอลชนิด Bit-Oriented หมายความว่า เป็นการส่งในลักษณะของ บิตสตรีม(Bit Stream) จากนั้นจึงจะแบ่งบิตข้อมูลในลักษณะของ ฟิลด์ (Field) นั่นคือเป็นโปรโตคอลที่สามารถส่งข้อมูลทั้งเฟรมในลักษณะของบิตสตรีม (แต่ในทางปฏิบัติก็ยังทำการส่งในลักษณะของไบต์)ซึ่งเป็นมาตรฐาน ISO โดยลักษณะของเฟรมของ HDLC โปรโตคอลจะมีลักษณะดังรูปที่ 2.2

แฟลก	แอดเดรส	ควบคุม	ข้อมูลที่ต้องการส่งจริง	ส่วนตรวจสอบความผิดพลาด
8 บิต	8 บิต	8 บิต	ตามความต้องการ	16 บิต

รูปที่ 2.2 แสดงรายละเอียดภายในเฟรมข้อมูลของ HDLC โปรโตคอล

องค์ประกอบภายในเฟรมของ HDLC โปรโตคอล

1) ส่วนของแฟลก จะมีขนาด 8 บิต มีรูปแบบ เป็น 01111110 ซึ่งการที่ส่วนของแฟลกมีรูปแบบดังนี้ทำให้รูปแบบของข้อมูลที่ตามหลังมาไม่สามารถมีรูปแบบเหมือนกับแฟลกได้ เพราะถ้าเมื่อใดก็ตามที่ข้อมูลที่ตามหลังส่วนของแฟลกมีรูปแบบเหมือนกันจะทำให้เกิดการเริ่มต้นของชุดข้อมูลชุดใหม่ทันที
เราแก้ปัญหาโดยวิธีการเพิ่มบิต " 0 " (Zero insertion) ลงไป ในทุกๆชุดของข้อมูลที่มีบิต " 1 " ติดกันทุกๆ 5 บิต ในส่วนของ แอดเดรส, ควบคุม, ข้อมูลที่ต้องการส่งจริง, และส่วนตรวจสอบการส่ง

2) ส่วนแอดเดรส เป็นส่วนที่จะบอกถึงสถานีที่ต้องการที่จะส่งข้อมูลต่างๆ ไปโดยการอ้างแอดเดรส อาจจะมีอยู่ หนึ่งแอดเดรสที่ถูกกำหนดให้เป็นการส่งในลักษณะของการส่งไปยังสถานีลูกข่ายทุกสถานี
(Broadcasting Address)

3) ส่วนควบคุม เป็นการบอกถึงชนิดของ เฟรม และจำนวนของเฟรมที่ทำการส่งและจำนวนของเฟรมที่ได้รับ และกำหนดรูปแบบของข้อมูลที่ต้องการส่งจริงซึ่งสามารถระบุชนิดของเฟรมได้ดังนี้คือ

1	2	3	4	5	6	7	8
0	N(S)			P/F	N(R)		

รูปที่ 2.3 ฟิลด์ส่วนควบคุมที่ระบุเป็นเฟรมข้อมูล

จากรูปที่ 2.3 ฟิลด์ส่วนควบคุมที่ระบุว่าเฟรมที่ส่งมาเป็นเฟรมข้อมูล (Information frame)

- เฟรมข้อมูล (Information frame) จะเป็นเฟรมของการรับ-ส่งข้อมูลแบบปกติ โดยบิตที่ 1 จะมีค่าเป็น ศูนย์, N(S) การบอกจำนวนของเฟรมที่จะส่งออกไป, N(R) จำนวนของเฟรมที่รับได้ ส่วน P/F แฟลก บิตที่ 5 จะเป็นการส่งบิต เพื่อร้องขอการตอบกลับ ซึ่งเมื่อมีการส่งแฟลกบิต P จากสถานีแม่ข่าย แล้วจะไม่สามารถส่งซ้ำสำหรับสถานีลูกข่ายเดิมได้ ต้องรอให้มีการตอบรับแฟลกบิต F จากสถานีลูกข่ายที่ถูกร้องขอการตอบรับก่อน จึงจะสามารถส่งแฟลกบิต P ซ้ำได้สำหรับสถานีลูกข่ายเลขที่เดิม ในขณะเดียวกัน สถานีที่ไม่ได้รับแฟลกบิต P ก่อนก็ไม่สามารถส่งแฟลกบิต F ได้

1	2	3	4	5	6	7	8
1	0	S		P/F	N(R)		

รูปที่ 2.4 ฟิลด์ส่วนควบคุมที่ระบุเป็นเฟรมซูเปอร์ไวซอรี

จากรูปที่ 2.4 ฟิลด์ส่วนควบคุมที่ระบุว่าเฟรมที่ส่งมาเป็นเฟรมซูเปอร์ไวซอรี (Supervisory frame)

- เฟรมซูเปอร์ไวซอรี (Supervisory frame) บิตที่ 1 และ 2 จะเป็นบิตที่บอกว่าเฟรมที่ส่งมาเป็นเฟรมซูเปอร์ไวซอรี โดยบิตที่ 1 จะมีค่าเป็น " 1 " และ บิตที่ 2 จะมีค่าเป็น " 0 " จะแบ่งเป็นการควบคุมของเฟรมชนิดนี้ออกเป็น 4 แบบ โดยการใช้ 2 บิต คือบิตที่ 3 และบิตที่ 4 เป็นบิตแบ่งชนิดของเฟรม

1. RR (Receive Ready) เป็นชนิดของเฟรมที่ใช้เพื่อร้องขอการตอบรับ
2. RNR (Receive Not Ready) เป็นชนิดของเฟรมที่ใช้เมื่อมีการเกิด Time out ขึ้นก็จะมีการส่งเฟรม RNR ซ้ำออกไปเพื่อ แต่จะระบุเป็น Receive Not Ready เฟรม
3. REJ (Reject) เป็นชนิดของเฟรมที่ร้องขอให้มีการส่งเฟรมข้อมูลซ้ำ โดยส่งซ้ำตั้งแต่เฟรมที่มีปัญหาเป็นต้นไป
4. SREJ (Selective Reject) เป็นชนิดของเฟรมที่ร้องขอให้มีการส่งเฉพาะเฟรมที่มีปัญหาเท่านั้น

1	2	3	4	5	6	7	8
1	1	M		P/F	M		

รูปที่ 2.5 ฟิลด์ส่วนควบคุมที่ระบุเป็นเฟรมไม่ระบุจำนวน

- เฟรมไม่ระบุจำนวน (Unnumbered frame) บิตที่ 1 และ 2 จะเป็นบิตที่บอกว่าเฟรมที่ส่งมาเป็นเฟรมไม่ระบุจำนวน โดยบิตที่ 1 จะมีค่าเป็น " 1 " และ บิตที่ 2 จะมีค่าเป็น " 1 " เป็นเฟรมคำสั่งสำหรับผู้ซ่อมบำรุงระบบ โดยจะไม่ใช้ในเวลาการทำงานปกติ

4) ส่วนของข้อมูลที่ต้องการส่งจริง ซึ่งสามารถส่งความยาวของข้อมูลได้ตามความต้องการ แต่รูปแบบของข้อมูลขึ้นอยู่กับส่วนควบคุม

5) ส่วนตรวจสอบความผิดพลาดของการส่ง เป็นส่วนในการตรวจสอบความผิดพลาดของข้อมูลในการส่ง โดยใช้รหัสโพลิโนเมียล โดย CCITT-16 ซึ่งการตรวจสอบนี้จะตรวจสอบ ส่วนแอดเดรส, ส่วนควบคุม และส่วนของข้อมูลที่ส่ง จะเห็นว่าการตรวจสอบของ HDLC จะเป็นการตรวจสอบแบบ Automatic Repeat ReQuest (ARQ) ซึ่งเป็นการตรวจสอบความผิดพลาดและร้องขอการส่งข้อมูลให้ใหม่จากสถานีต้นทาง

ข้อด้อยของ โปรโตคอล HDLC [1][5]

1. จะเห็นว่า โปรโตคอล HDLC มีข้อยืดหยุ่นค่อนข้างมากนั่นคือ ผู้ผลิตสามารถที่จะออกแบบการส่งข้อมูลและการควบคุมได้กว้าง ทำให้อุปกรณ์บางอย่างที่ระบุว่าใช้ โปรโตคอล HDLC อาจจะไม่สามารถติดต่อสื่อสารกันได้

2. การมี Zero insertion ซึ่งเป็นการแก้ปัญหาสำหรับการส่ง บิตที่มีค่า " 1 " ติดกัน 5 บิต เพื่อไม่ให้ซ้ำกับแฟลก ทำให้ไม่ค่อยสะดวกนักในเชิงปฏิบัติ เนื่องจากอาจมีผู้ผลิตบางรายใช้ ใน Physical layer ในขณะที่ผู้ผลิตบางรายใช้ใน Data Layer

3. การมี Zero insertion จะมีปัญหาในกรณีของการเกิด การส่งข้อมูลผิดพลาด ซึ่งจะทำให้ FCS (Frame Check Sequence) เกิดการผิดพลาดในการตรวจจับข้อมูล

4. ส่วนตรวจสอบความผิดพลาดในการส่ง ไม่มีการตรวจสอบการผิดพลาดของ FLAG เริ่มต้น และ FLAG สิ้นสุด ทำให้ถ้ามีการผิดพลาดใน FLAG ทั้งสองจะทำให้เสียข้อมูลทั้งเฟรม

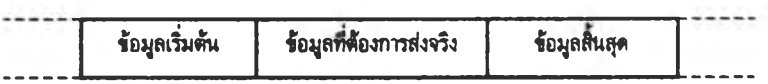
2.3.2 IEEE Std 999-1992 โปรโตคอล [1]

สถาบัน IEEE แนะนำให้ใช้โปรโตคอลนี้เพื่อใช้กับระบบสเกดาโดยเฉพาะ มีรายละเอียดดังต่อไปนี้

รูปแบบของข้อมูลในการสื่อสารระหว่างสถานีแม่ข่ายและสถานีลูกข่าย

รูปแบบและโครงสร้างข้อมูลที่จะกล่าวถึงต่อไปนี้ ได้ใช้หลักการตามเอกสารแนะนำของ IEEE สำหรับระบบควบคุมและติดตามผลข้อมูล

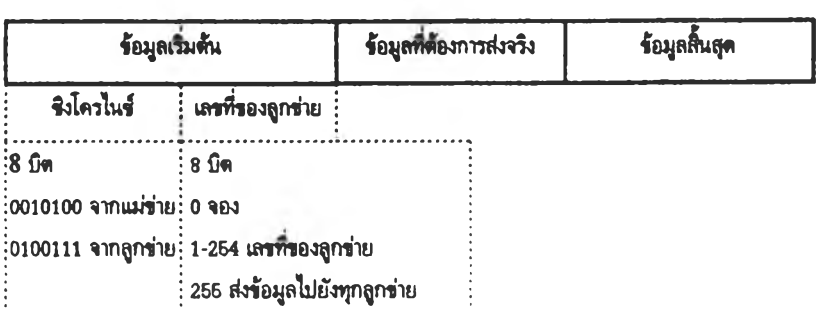
โครงสร้างโดยทั่วไปของข้อมูลที่ติดต่อระหว่างแม่ข่ายและลูกข่ายผ่านทางสื่อต่างๆ ดังที่กล่าวไว้เบื้องต้น จะมีรูปแบบดังรูปที่ 2.6



รูปที่ 2.6 โครงสร้างของลำดับการส่งข้อมูล IEEE std999-1992

การสื่อสารระหว่างแม่ข่ายกับลูกข่ายจะใช้การส่งข้อมูลแบบอนุกรม และจะประกอบด้วยข้อมูล 3 ส่วน คือ ส่วนข้อมูลเริ่มต้น, ส่วนของข้อมูลที่ต้องการส่งจริง และส่วนข้อมูลสิ้นสุด

ข้อมูลเริ่มต้น

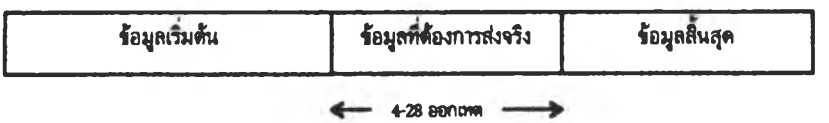


รูปที่ 2.7 รายละเอียดในส่วนของข้อมูลเริ่มต้น

ข้อมูลเริ่มต้นจะประกอบด้วย 2 ส่วนคือ

- ส่วนซิงโครไนซ์ มีขนาด 8 บิต มีขึ้นเพื่อให้ภาครับของทั้งแม่ข่ายและลูกข่ายได้ รับรู้ว่าข้อมูลที่จะตามมาเป็นข้อมูลชุดใหม่ที่ส่งมาจากแม่ข่ายหรือลูกข่าย นอกจากนี้ยังทำหน้าที่แยกระหว่างการส่งจากสถานีแม่ข่ายหรือสถานีลูกข่ายอีกด้วย
- ส่วนเลขที่ของลูกข่าย มีขนาด 8 บิต มีหน้าที่บอกเลขที่ของลูกข่ายที่กำลังติดต่ออยู่กับแม่ข่ายโดยกำหนดให้จำนวนลูกข่ายมีได้ตั้งแต่ 1-254 สถานี สำหรับเลขที่ 255 จะเป็นการส่งจากสถานีแม่ข่ายเพื่อกระจายข้อมูลไปยังทุกสถานีลูกข่ายโดยไม่ต้องมีการตอบกลับของสถานีลูกข่าย สำหรับเลขที่ 0 เป็นการจองหมายเลขสำหรับการใช้งานในอนาคต

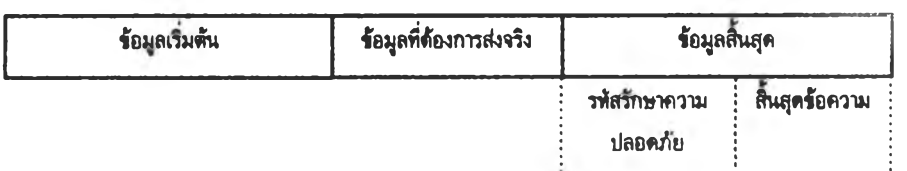
ข้อมูลที่ต้องการส่งจริง



รูปที่ 2.8 จำนวนออกเทตในส่วนของข้อมูลที่ต้องการส่ง

ข้อมูลที่ต้องการส่งจะแบ่งเป็นส่วนย่อยขนาด 8 บิต มีตั้งแต่ 4- 28 ออกเทต (Octet)
โดยที่ ออกเทต เป็นหน่วยของขนาดของข้อมูลความยาว 8 บิต

ข้อมูลสิ้นสุด

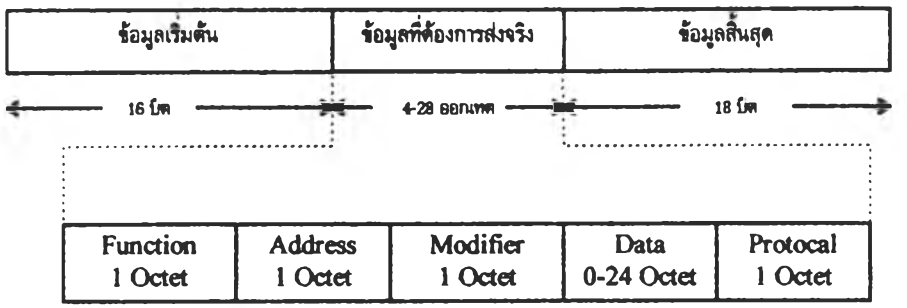


รูปที่ 2.9 ส่วนประกอบในส่วนของข้อมูลสิ้นสุด

- รหัสรักษาความปลอดภัย มีขนาด 16 บิต เป็นส่วนที่คำนวณข้อมูลที่ส่งมาเบื้องต้นทั้งหมด ยกเว้นบิตแรก(ของ)ส่วนซิงโครไนซ์ เป็นการป้องกันการเกิดข้อผิดพลาดของการส่งข้อมูล รวมทั้งการเกิดซิงก์สลิป (Sync Slip) คือการที่ส่วนรับสัญญาณไม่สามารถตรวจจับส่วนของซิงโครไนซ์ ที่ระบุส่วนเริ่มต้นของเฟรมได้
- ส่วนสิ้นสุดข้อความ การส่งการสิ้นสุดข้อความจะใช้ขนาด 2 บิตสุดท้ายซึ่งมีลอจิกเป็น " 1 " ทั้งสองบิต

รูปแบบของข้อมูลที่ต้องการส่งจริง (Information Field Format)

- จากสถานีแม่ข่าย ไป สถานีลูกข่าย



รูปที่ 2.10 รายละเอียดในส่วนของข้อมูลที่ส่งจากสถานีแม่ข่ายไปสถานีลูกข่าย

Function : เป็นส่วนที่จะกำหนดหน้าที่ของคำสั่งหรือการส่งข้อมูลที่สถานีแม่ข่ายกำหนดต่อสถานีลูกข่ายซึ่งสามารถกำหนดได้ถึง 256 คำสั่งหรือหน้าที่

Address : เป็นส่วนที่กำหนดตำแหน่งภายในของข้อมูล ซึ่งสามารถกำหนดได้ 256 ตำแหน่ง

Modifier : เป็นส่วนที่กำหนดจำนวนออกเทตของข้อมูล หรือกำหนดจำนวนของข้อมูลที่จะทำการส่ง

Data : เป็นส่วนของการส่งข้อมูล โดยส่งได้ถึง 24 ออกเทต

Protocol : มีขนาด 8 บิต เป็นตัวกำหนดการรับส่งข้อมูลในเงื่อนไขต่างๆ ของการปฏิบัติงานระหว่างสถานีแม่ข่ายและสถานีลูกข่าย

บิต 0 เป็นบิตที่ทำการรีเซ็ตแฟลกบิตของสถานีลูกข่าย ซึ่งเป็นบิตที่สถานีลูกข่ายจะเซ็ตขึ้นเมื่อมีปัญหาเกิดขึ้นที่สถานีลูกข่าย

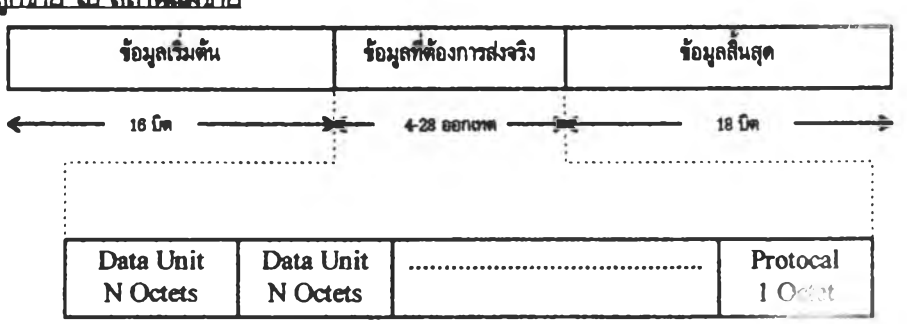
บิต 1 เป็นบิตที่ทำการรีเซ็ตแฟลกบิตของสถานีลูกข่าย ซึ่งเป็นบิตที่สถานีลูกข่ายจะเซ็ตขึ้นเมื่อเกิดการผิดพลาดในการส่งข้อมูลจากสถานีแม่ข่าย

บิต 2 เป็นบิตที่ทำการรีเซ็ตแฟลกบิตของสถานีลูกข่าย ซึ่งเป็นบิตที่สถานีลูกข่ายจะเซ็ตขึ้นเมื่อได้รับข้อมูลในแบบการกระจายข้อมูลจากสถานีแม่ข่าย

บิต 3 เป็นบิตที่ทำการเซ็ตจากสถานีแม่ข่ายเพื่อบอกให้สถานีลูกข่ายทราบว่าข้อมูลที่ส่งไปเป็นข้อมูลที่มีความยาวของข้อมูล

บิต 4-7 เป็นบิตที่ยังไม่ได้กำหนดหน้าที่ ซึ่งผู้ใช้งานอาจนำไปกำหนดหน้าที่ที่ต้องการเพิ่มขึ้นได้

- จากสถานีลูกข่าย ไป สถานีแม่ข่าย



รูปที่ 2.11 รายละเอียดในส่วนของข้อมูลที่ส่งจากสถานีลูกข่ายไปสถานีแม่ข่าย

Data Unit : หน่วยของข้อมูลผู้ใช้สามารถกำหนดจำนวนของออกเทต ตามความสะดวกของผู้ใช้งาน แต่จะต้องไม่น้อยกว่า 3 ออกเทต และไม่มากกว่า 27 ออกเทต

Protocol : มีขนาด 8 บิต มีลักษณะการใช้งานตามอธิบายด้านล่าง

บิต 0 : แฟลกบิตที่ถูกเซ็ตโดยสถานีลูกข่ายเมื่อ เกิดการทำคำสั่งรีสตาร์ต หรือ เมื่อตรวจสอบพบปัญหาจากอุปกรณ์ และแฟลกนี้จะถูกรีเซ็ตโดยสถานีแม่ข่าย

คำสั่งรีสตาร์ตจะกระทำเมื่อ

- เกิดการหยุดจ่ายไฟฟ้ากระทันหัน
- ได้รับคำสั่งรีสตาร์ตจากสถานีแม่ข่าย
- ได้รับคำสั่งโดยตรงจากผู้ปฏิบัติงาน
- รีเซ็ตตัวเอง

ปัญหาจากอุปกรณ์ที่จะทำให้เกิดการเซ็ต

- บัฟเฟอร์ที่ใช้เก็บข้อมูลของเหตุการณ์ ถูกใช้งานไป 75%
- บัฟเฟอร์ที่ใช้เก็บข้อมูลรายงานแอนะลอก ถูกใช้งานไป 75%
- สัญญาณแอนะลอกอินพุตมีปัญหา
- สัญญาณคอนโทรลเอาต์พุตมีปัญหา

บิต 1 : แฟลกบิตเกิดจากการสื่อสารผิดพลาด เกิดจากการที่การส่งสัญญาณสื่อสารข้อมูลผิดพลาดจากสถานีแม่ข่าย

บิต 2 : แฟลกบิตนี้จะถูกเซ็ตเมื่อสถานีลูกข่ายได้รับคำสั่งการกระจายข้อมูลจากสถานีแม่ข่าย โดยคุณสมบัติของคำสั่งนี้มีเพื่อ - Analog Freeze Command

- Accumulator Freeze Command
- คำสั่งซิงโครไนซ์ฐานเวลา

บิต 3 : แฟลกบิตนี้จะถูกเซ็ตและรีเซ็ตโดยสถานีลูกข่ายเท่านั้น เนื่องจากสถานีลูกข่ายคาดว่าจะมีการส่งข้อมูลที่มีความยาว

บิต 4-7 : ไม่ถูกกำหนด ซึ่งผู้ใช้งานอาจใช้ประโยชน์จากบิตเหล่านี้ได้

ข้อเปรียบเทียบระหว่าง HDLC โปรโตคอล และ IEEE Std999-1992 โปรโตคอลในเบื้องต้น

โดยกำหนดเงื่อนไขดังนี้

การส่งจากสถานีแม่ข่ายไปยังสถานีลูกข่าย จะไม่ส่งข้อมูลใดๆแต่จะส่งแค่สัญญาณควบคุมต่างๆเท่านั้น

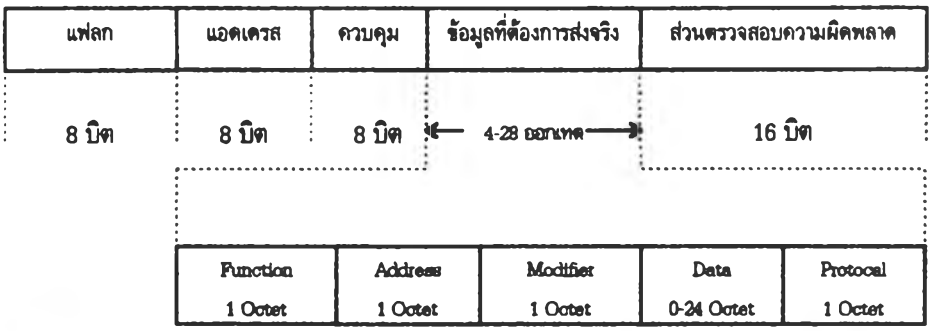
การส่งจากสถานีลูกข่ายตอบรับสถานีแม่ข่าย จะส่งข้อมูลระหว่าง 3 - 27 ออกเทต

1) ความสามารถในการส่งผ่านข้อมูล (Transaction Throughput Capacity)

<u>HDLC โปรโตคอล</u>

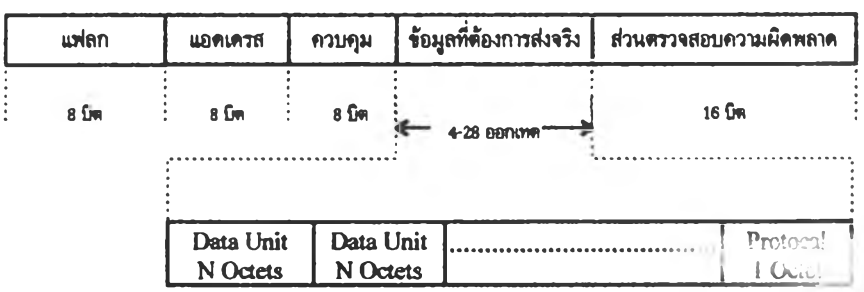
เรากำหนดให้จำนวนข้อมูลในการส่งจากสถานีลูกข่ายมายังสถานีแม่ข่ายมีจำนวนเป็น D ออกเทต

การติดต่อของ HDLC สำหรับระบบสเกดาจะใช้ 9 ออกเทต ซึ่งจะรวม ส่วนของแฟลก, แอดเดรส, ส่วนควบคุม, ข้อมูลที่ต้องการส่งจริง(ยกเว้นข้อมูล) และส่วนตรวจสอบความผิดพลาด ในส่วนของการส่งจากสถานีแม่ข่าย มายังสถานีลูกข่าย ดังรูปที่ 2.12



รูปที่ 2.12 รายละเอียดส่วนข้อมูลที่ส่งจริงของ HDLC จากสถานีแม่ข่าย ไป สถานีลูกข่าย

และอีก 6 ออกเทตซึ่งจะมี แฟลก, แอดเดรส, ส่วนควบคุม, ข้อมูลที่ต้องการส่งจริง(เฉพาะ โปรโตคอล) และส่วนตรวจสอบความผิดพลาด รวมกับข้อมูลจำนวน D ออกเทต สำหรับสถานีลูกข่ายกลับมายังสถานีแม่ข่าย ดังรูปที่ 2.13



รูปที่ 2.13 รายละเอียดส่วนข้อมูลที่ส่งจริงของ HDLC จากสถานีลูกข่าย ไป สถานีแม่ข่าย

จะเห็นว่าความสามารถในการส่งผ่านข้อมูลของ HDLC โปรโตคอลมีค่าเท่ากับ

(D) / (D + 9 + 6) เมื่อ D มีค่าตั้งแต่ 3 - 27 ออกเทต

IEEE Std 999-1992 โปรโตคอล

จากรูปที่ 2.10 เป็นการส่งจากสถานีแม่ข่ายไปยังสถานีลูกข่าย เมื่อไม่มีการส่งข้อมูล จะมีค่าเท่ากับ 8 ออกเทต และจากรูปที่ 2.11 เป็นการตอบกลับของสถานีลูกข่ายกลับมาสถานีแม่ข่าย มีค่าเท่ากับ 5 รวมกับ D ออกเทต โดยที่ D เป็นหน่วยของข้อมูลที่จำนวนตั้งแต่ 3 - 27 ออกเทต ซึ่งเราจะได้ความสามารถในการส่งผ่านข้อมูลของ IEEE Std 999-1992 โปรโตคอล มีค่าเท่ากับ

(D) / (D + 8 + 5) เมื่อ D มีค่าตั้งแต่ 3 - 27 ออกเทต

เมื่อเปรียบเทียบโปรโตคอลทั้ง 2 ชนิดจะได้ดังนี้

ที่ D = 3 ออกเทต จะได้ HDLC โปรโตคอล มีความสามารถในการส่งผ่านข้อมูล = 16.67 %

IEEE Std 999-1992 โปรโตคอล มีความสามารถในการส่งผ่านข้อมูล = 18.75 %

ที่ D = 27 ออกเทต HDLC โปรโตคอล มีความสามารถในการส่งผ่านข้อมูล = 64.28 %

IEEE Std 999-1992 โปรโตคอล มีความสามารถในการส่งผ่านข้อมูล = 67.5 %

จะเห็นว่าความสามารถในการส่งผ่านข้อมูลของ IEEE Std 999-1992 โปรโตคอลสูงกว่า HDLC โปรโตคอลที่การส่งข้อมูลระหว่าง 3-27 ออกเทต

2) เปรียบเทียบคุณสมบัติการส่งในส่วนของการซิงโครไนซ์ของการส่งแต่ละครั้ง

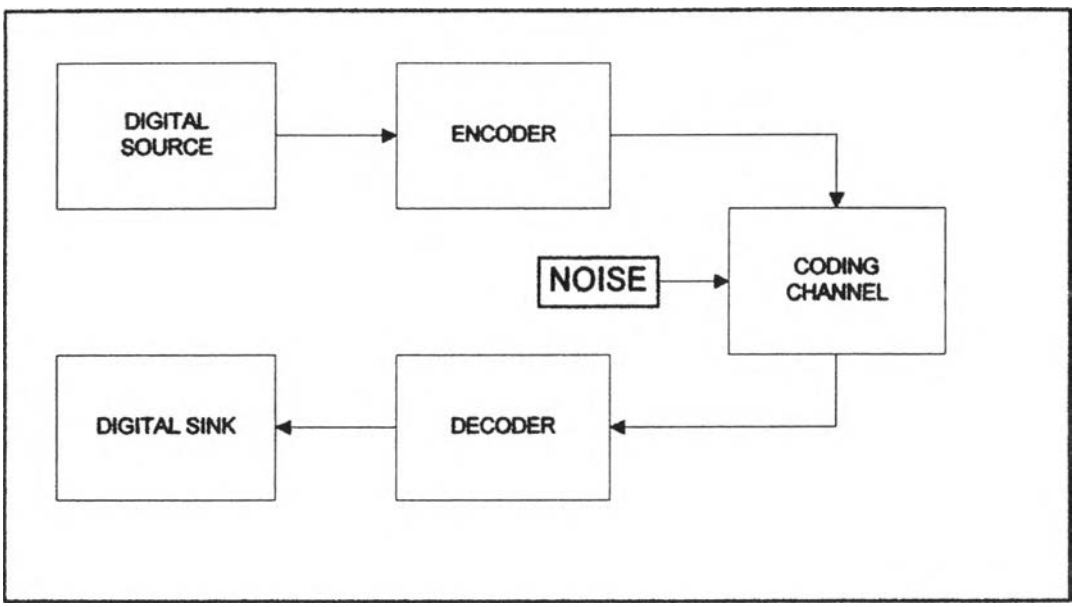
จะเห็นว่า HDLC โปรโตคอลมีคุณสมบัติการส่งในส่วนของการซิงโครไนซ์ด้อยกว่า IEEE Std 999-1992 โปรโตคอล เนื่องจากส่วนตรวจสอบความผิดพลาดของข้อความของ HDLC โปรโตคอล ไม่ได้ตรวจสอบการผิดพลาดของส่วนแฟลกการซิงโครไนซ์ ในขณะที่ IEEE Std 999-1992 ส่วนรหัสรักษาความปลอดภัยมีการตรวจสอบความผิดพลาดของส่วนการซิงโครไนซ์ ดังนั้น ถ้ามีการผิดพลาดเกิดขึ้นที่ส่วนแฟลกการซิงโครไนซ์ของ HDLC โปรโตคอล จะทำให้การส่งข้อความครั้งนั้นผิดพลาดทั้งหมดต้องมีการส่งใหม่ทั้งข้อความ

ข้อเปรียบเทียบของโปรโตคอล IEEE std999-1992 กับ HDLC สำหรับการใช้งานในระบบสเกดา		
รายละเอียด	HDLC	IEEE std999-1992
1. ความสามารถในการส่งผ่านข้อมูล (Throughput)	เปอร์เซ็นต์ ต่ำกว่า IEEE std 999-1992	เปอร์เซ็นต์ ดีกว่า HDLC เล็กน้อย
2. มีการแทรกบิต 0 (Zero Insertion) เพื่อป้องกันการซ้ำแฟลกเริ่มต้นข้อมูล	ต้องมีการแทรกบิต 0	ไม่มี
3. การตรวจสอบแฟลกเริ่มต้นของเฟรม	ไม่มีการตรวจสอบแฟลกเริ่มต้นข้อมูล ซึ่งอาจทำให้เกิดซิงก์สลิปได้	รวมการตร... พลาดใน...
4. การตรวจสอบความผิดพลาดในการส่งข้อมูล	ใช้แบบ ARQ (Automatic Repeat Request)	ใช้แบบ FEC (Forward Error Correction)

ตารางที่ 2.1 ข้อเปรียบเทียบระหว่าง HDLC กับ IEEE std999-1992

2.4 การตรวจสอบและการแก้ไขความผิดพลาดของการส่งข้อมูล (ERROR CONTROL AND ERROR CORRECTION) [1][2][6][13]

โดยพื้นฐานความต้องการในเบื้องต้นจึงทำให้ต้องเพิ่มระบบเข้ารหัสและระบบถอดรหัส เข้าไปในระบบการส่งสัญญาณข้อมูลดิจิตอล ดังมีหลักการพื้นฐานดังนี้



รูปที่ 2.14 บล็อกไดอะแกรมพื้นฐานของ ระบบเข้ารหัสและถอดรหัส

การควบคุมความผิดพลาดของการส่งข้อมูลมี 2 แบบใหญ่คือ

2.4.1 FORWARD ERROR CORRECTION (FEC)

เป็นการควบคุมความผิดพลาดของการส่งข้อมูลแบบทางเดียว ดังรูปที่ 2.14 ซึ่งระบบจะทำการส่งข้อมูลที่เข้ารหัส ไปยัง สถานีรับข้อมูล จากนั้นจะทำการถอดรหัส และ แก้ไขในกรณีที่มีการผิดพลาดในการส่งข้อมูล และส่งต่อไปยัง อุปกรณ์รับ รูปแบบการควบคุมแบบนี้เป็นรูปแบบที่ใช้กันมากในปัจจุบัน แม้ว่าช่องการรับส่งข้อมูล (Channel) จะไม่เป็นการส่งแบบทางเดียวก็ตาม โดยที่ โปรโตคอล IEEE std 999-1992 จะใช้การควบคุมความผิดพลาดในการส่งข้อมูลแบบนี้

2.4.2 AUTOMATIC REPEAT REQUEST (ARQ)

เป็นการควบคุมความผิดพลาดของการส่งข้อมูลแบบ สองทาง คือมีการร้องขอการส่งข้อมูลซ้ำในกรณีที่เกิดการผิดพลาดในการส่งข้อมูลครั้งแรก ระบบการควบคุมแบบนี้มี 2 รูปแบบคือ

2.4.2.1 STOP AND WAIT ARQ คือ เมื่อสถานีส่งข้อมูลที่ทำการเข้ารหัสแล้ว สถานีส่งจะรอรับคำสั่งตอบกลับจาก สถานีรับ ว่าข้อมูลที่ส่งมีข้อผิดพลาดหรือไม่ ในกรณีที่ไม่มีการผิดพลาดของข้อมูล จะเป็น

Positive Acknowledgment (ACK) ถ้าเกิดการผิดพลาดขึ้นในการส่งข้อมูล สถานีรับจะส่ง คำสั่งแบบ Negative Acknowledgment (NAK) ซึ่งสถานีส่งเมื่อได้รับคำสั่งแล้วจะทำการส่งซ้ำอีกครั้งและรอรับคำตอบซึ่งอาจจำเป็นต้องมีการส่งซ้ำหลายครั้ง

2.4.2.2 Continuous ARQ คือ สถานีส่งจะส่งข้อมูลที่เข้ารหัสอย่างต่อเนื่อง ในขณะที่สถานีรับก็จะส่งคำสั่ง Acknowledgment อย่างต่อเนื่องเช่นกัน จนกว่าสถานีส่ง จะได้รับข้อมูล NAK สถานีส่งก็จะทำการเริ่มต้นในการส่งใหม่ อาจเป็นการส่งซ้ำเฉพาะที่เกิดข้อผิดพลาดเป็นต้นไป(go-back N ARQ)หรือ ส่งซ้ำเฉพาะส่วนที่เกิดผิดพลาด (selective repeat ARQ) ก็ขึ้นอยู่กับรูปแบบ

การส่งข้อมูลในแบบนี้จะมีราคาแพงกว่าแบบแรก แต่เหมาะสำหรับการสื่อสารซึ่งมีเวลาการรับ-ส่งข้อมูลที่นาน ซึ่งโปรโตคอลของ HDLC และอีกหลายๆชนิด จะใช้การตรวจสอบข้อมูลผิดพลาดแบบนี้

เราจะกล่าวถึงรายละเอียดของการควบคุมความผิดพลาดของการส่งข้อมูลในแบบของ FEC (FORWORD ERROR CORRECTION) เท่านั้นเพราะมีรายละเอียดและรูปแบบในการเลือกชนิดของ FECในการใช้งานมาก

เราสามารถแบ่งชนิดของ FEC ได้เป็น 2 ลักษณะใหญ่คือ

1. รหัสบล็อก (BLOCK CODE) เป็นการเข้ารหัสแบบบล็อก จะแบ่งข้อมูลที่จะทำการเข้ารหัสออกเป็นบล็อก ซึ่งเราจะเรียกว่าเป็น บล็อกข้อมูล (MESSAGE BLOCK) ซึ่งจะมีทั้งหมด k บิต เราจะเห็นว่าเราสามารถมีข้อมูลที่แตกต่างกันภายในบล็อกข้อมูล ได้ถึง 2^k ข้อมูลจากข้อมูลที่แตกต่างกันนี้เมื่อเราจะต้องเข้ารหัสเพื่อหารหัส (CODE WORD) ที่แตกต่างกันถึง 2^k รหัสเป็นอย่างน้อย ซึ่งค่าของ 2^k รหัสเท่ากับความยาว n จะถูกเรียกว่า รหัสบล็อก (n,k)

2. รหัสคอนโวลูชันแนล (CONVOLUTIONAL CODE) เป็นการเข้ารหัสโดยที่จะใช้ ข้อมูล k บิตในข้อมูลที่ต้องการจะส่งเช่นกัน การเข้ารหัสของคอนโวลูชันแนล จะมีลักษณะที่คล้ายกับ รหัสบล็อก แต่การเข้ารหัสของคอนโวลูชันแนล จะมีข้อมูลของบล็อกที่ส่งไปก่อนหน้ามาเป็นเงื่อนไขในการเข้ารหัสครั้งปัจจุบัน เข้ารหัสจะต้องมีหน่วยความจำ ของข้อมูลบล็อกที่ส่งไปแล้วเก็บอยู่ด้วย โดยที่เราแทน m เป็นการเก็บข้อมูลออเดอร์ที่ m เราเรียกได้โดยที่ k เท่ากับ จำนวนอินพุตบิต , n เท่ากับ เอาต์พุตที่เข้ารหัสของหน่วยความจำในการเก็บข้อมูลออเดอร์ที่ m ว่า รหัสคอนโวลูชันแนล (n,k,m)

รหัสบล็อกเชิงเส้น (LINEAR BLOCK CODE)

เราพิจารณาการกระทำใดๆ ในเซ็ตของไบนารีโดยกำหนดให้เซ็ต G มีสมาชิกเป็นไบนารีและเป็นเซ็ตที่จำกัด และการกระทำใดๆ ระหว่างสมาชิกภายในเซ็ต G จะต้องอยู่ภายในเซ็ต G

โอเปอแรนด์สำหรับไบนารีเรากำหนดให้มี 2 ตัวคือ

การบวก

+	0	1
0	0	1
1	1	0

การคูณ

●	0	1
0	0	0
1	0	1

- Primitive Polynomial P(x)
- GALOIS FIELD GF(2^m)
- Property of GALOIS FIELD
- Minimal Polynomial $\varnothing(X)$

polynomial p(x) ซึ่งมีกำลังสูงสุด = m($m \geq 1$)และไม่สามารถหารด้วย polynomial อื่นๆ นอกจากตัวเองได้ลงตัวในขณะเดียวกันต้องเป็น polynomial ที่หาร X^n+1 โดยที่ $n = 2^m-1$ เราจะเรียกว่า primitive polynomial

GF(2^m) เป็นเซ็ตที่มีจำนวนสมาชิกทั้งหมด 2^m โดยที่ $m>1$ จากเซ็ตของไบนารีในเบื้องต้นที่มีสมาชิก 0 และ 1 จาก GF(2) เรากำหนดสัญลักษณ์ α และเราสามารถแสดงโอเปอแรนด์สำหรับ α ได้ดังนี้

การคูณ

$0.0=0$

$0.1=1.0=1$

$1.1=1$

$0.\alpha=\alpha.0=0$

$1.\alpha=\alpha.1=\alpha$

$\alpha^2=\alpha.\alpha$

$\alpha^3=\alpha.\alpha.\alpha$

$0.\alpha^j=\alpha^j.0=0$

$1.\alpha^j=\alpha^j.1=\alpha^j$

$\alpha^i.\alpha^j=\alpha^j.\alpha^i=\alpha^{i+j}$

จากทฤษฎีของ primitive polynomial p(x)

$X^{2m-1}+1 = q(x)\ p(x)$

เราสมมุติให้ $p(x) = 0$

ดังนั้น $\alpha^{2m-1} + 1 = q(\alpha).p(\alpha) = 0$

จากการโอเปอร์แรนด์ + จะได้ $\alpha^{2m-1} = 1$ เพราะฉะนั้นจากเงื่อนไข p(x) = 0 เราจะได้เซ็ต F = { 0, 1, $\alpha, \alpha^2, \ldots \alpha^{2m-2}$ } จากเซ็ต F และเงื่อนไข GF(2^m) โดยที่ $p(\alpha) = 0$ จะทำให้เราได้ข้อจำกัดของ polynomial สำหรับโอเปอร์แรนด์การบวกคือ

$0 + 0 = 0$

$0 + \alpha^i = \alpha^i + 0 = \alpha^i$

$\alpha^i + \alpha^i = \alpha^i + 0 = \alpha^i$

$\alpha^i + \alpha^i = 0$

คุณสมบัติพื้นฐานของ GF(2^m)

$f(x)$ เป็นฟังก์ชั่น polynomial พร้อม 2^m-1 เป็นจำนวนสมาชิกที่ไม่เป็นศูนย์ GF(2^m) และเป็นรากของฟังก์ชั่น $X^{2m-1}+1$ ทั้งหมด

$\phi(X)$ เป็น Minimal Polynomial ที่ไม่สามารถมีรากที่เป็น Polynomial อื่นใดได้ภายใน GF(2^m) ดังสมการ

$$\phi(X) = \prod_{i=0}^{e-1} (X+\beta^{2i})$$

โดยที่ e เป็นจำนวนเต็ม $\leq m$

$$\beta^{2e}=\beta$$

ทำให้เราสามารถหา Generator Polynomial $g(x) = \{ \phi_1(X), \phi_3(X), \ldots, \phi_{2t-1}(X) \}$

v = เวกเตอร์รหัส (code vector)

r = เวกเตอร์ภาครับ (receive vector)

d_{min} =minimum weight

minimum distance = d_{min} คือความสามารถในการตรวจจับ error pattern การที่ไม่สามารถตรวจจับ error ได้ หมายความว่าเกิด error จนกระทั่งทำให้ receive vector ที่ได้เหมือนกับ code vector อันอื่น

(n,k) รหัสเชิงเส้น (linear code) สามารถ detect ได้ถึง 2^n-2^k pattern

2^n-1 รูปแบบการผิดพลาด (error pattern) ไม่เป็น 0

2^k-1 nonzero code word

อาจจะมี 2^k-1 error pattern ที่มีเหมือนกับ 2^k-1 nonzero code word

ในกรณีที่ส่ง code word V และมี error ขึ้นทำให้เปลี่ยนเป็น W ซึ่ง W เป็น 1 ใน code word ที่ทำให้ Syndome = 0

ดังนั้นจะเห็นว่ามี 2^k-1 ที่ไม่สามารถตรวจจับความผิดพลาด ได้ ในกรณีที่ n มีค่ามากๆ $2^n-1 >> 2^k-1$

$t = (d_{min} -1)/2$

ความน่าจะเป็นของการถอดรหัสที่ไม่สามารถตรวจจับความผิดพลาดได้ จะกำหนดได้จาก Pu(E) เป็น probability of an undetected error

$$Pu(E) \le \sum_{i=t+1}^{n} \binom{n}{i} p^{i} (1-p)^{n-i}$$

รหัส BCH (BOSE CHAUDHURI and HOCQUENGHEM) [6]

เป็นรหัสของการตรวจสอบและแก้ไข การเกิดการผิดพลาดของข้อมูลที่เกิดหลายตำแหน่งแบบไซคลิกที่มีประสิทธิภาพสูง

กำหนดให้ทุกจำนวนเต็มบวก m ($m \ge 3$) และ t ($t < 2^{m-1}$) จะสามารถกำหนดตัวแปรอื่นๆ ได้ดังนี้

ความยาวของบล็อก $n = 2^{m} - 1$

จำนวนบิตของส่วนตรวจสอบ $n - k \le mt$

ระยะทางต่ำที่สุด (Minimum Distance) $d_{min} \ge 2t + 1$

เราจะสามารถกำหนดได้ว่า รหัสของเราสามารถที่จะแก้ไขบิตที่เกิดการผิดพลาดรวมกันได้ไม่เกินค่า t ในความยาวของบล็อกการส่งข้อมูลทั้งหมด n บิต เราเรียกว่า รหัส t error correcting BCH

สำหรับ BCH (255,239) โดยที่ t = 2 เราจะหาค่า Minimal Polynomial ได้ดังนี้

n = 255, k = 239, m = 8

$$\phi_1(X) = 1+X^2 + X^3 + X^4 + X^8$$

$$\phi_3(X) = 1+X + X^2 + X^4 + X^5 + X^6 + X^8$$

$$g(x) = \phi_1(X).\ \phi_3(X)$$

$$g(x) = 1+X + X^5 + X^6 + X^8 + X^9 + X^{10} + X^{11} + X^{13} + X^{14} + X^{16}$$

เพราะฉะนั้นเราสามารถหา Generator Polynomial เพื่อใช้สำหรับเป็นการ... ได้ โดย

$v(x) = u(x)g(x)$

v(x) = เวกเตอร์ที่เข้ารหัส (code vector)

u(x) = ข้อความที่จะทำการเข้ารหัส (information to be encoded)

g(x) = generator polynomial

ในกรณีที่ส่ง v(x) มายังส่วนของการรับสัญญาณจะทำให้เราได้ r(x) เวกเตอร์ภาครับ (receive vector) โดยที่

$r(x) = v(x) + e(x)$

e(x) = รูปแบบการผิดพลาด (error pattern)

ที่ส่วนของภาครับเราจะหาร r(x) ด้วย $\phi_1(X)$ และ $\phi_3(X)$ ซึ่งถ้า e(x) = 0 หมายถึงไม่มีความผิดพลาดเกิดขึ้น จะทำให้การหารไม่มีเศษของการหารเกิดขึ้น นั่นคือ r(x) = v(x)

เนื่องจาก $\phi_1(X)$. $\phi_3(X)$ เป็น minimal polynomial ของ g(x) แต่ถ้า e(x) $\neq$ 0 นั่นคือ r(x) = v(x) + e(x) เมื่อหาร r(x) ด้วย $\phi_1(X)$. $\phi_3(X)$ แล้วจะเหลือเศษเกิดขึ้นโดยเราจะนำเศษที่ได้นั้นมาคำนวณ เพื่อหาตำแหน่งของบิตที่ผิดพลาด

การหาตำแหน่งของบิตที่ผิดพลาด

เราสามารถหาตำแหน่งของบิตที่ผิดพลาดได้ โดยเมื่อ Receive Vecter เกิด e(x) error pattern ขึ้นโดยที่ e(x) มีค่าไม่เป็นศูนย์ และ S_j คือ Syndrome ดังสมการ

$$e(\alpha^j) = \sum_{i=1}^{v} Y_i X_i^j = S_j$$

Y_i คือ Magnitude of error เป็นสมาชิกของ GF(q)

X_i^j คือ ตำแหน่งของบิตผิดพลาด เป็นสมาชิกของ $GF(2^m)$

$e(\alpha^j) = S_j$

โดยที่ จำนวนของ S_j เท่ากับ $2t_0$

สมมุติให้ $v \leq t_0$ เป็นบิตผิดพลาดที่เกิดจริง เราจะได้สมการดังนี้

$$(X+X_1)(X+X_2)...(X+X_v) = X^v + \sigma_1 X^{v-1} + ... + \sigma_{v-1} X + \sigma_v = 0$$

นำ $Y_i X_i^j$ คูณสมการด้านขวาจะได้

$$Y_i X_i^{j+v} + \sigma_1 Y_i X_i^{j+v-1} + ... + \sigma_{v-1} Y_i X_i^{j+1} + \sigma_v Y_i X_i^j = 0$$

โดยที่ $1 \leq i \leq v$

ดังนั้นจะได้

$$S_{j+v} + S_{j+v-1} \sigma_1 + ... + S_{j+1} \sigma_{v-1} + S_j \sigma_v = 0$$

ขั้นตอนในการหาตำแหน่งของบิตที่ผิดพลาดกระทำได้ดังต่อไปนี้

1. คำนวณหา Syndrome S_j จาก Receive Vector
2. กำหนดค่าสูงสุดของการเกิดบิตผิดพลาด ในที่นี้เราให้ v = 2
3. คำนวณค่าของ σ_1 , .. ,σ_v จากสมการโดยแทนค่า Syndrome ที่หามาได้
4. นำค่าของ σ_1 , .. ,σ_v ที่คำนวณได้จากข้อ 3. มาแทนในสมการ

$$X^v + \sigma_1 X^{v-1} + ... + \sigma_{v-1} X + \sigma_v = 0$$

หาค่าของ X1,.., X^v

เมื่อเราได้ค่าของ X นั่นก็คือตำแหน่งของบิตที่ผิดพลาด