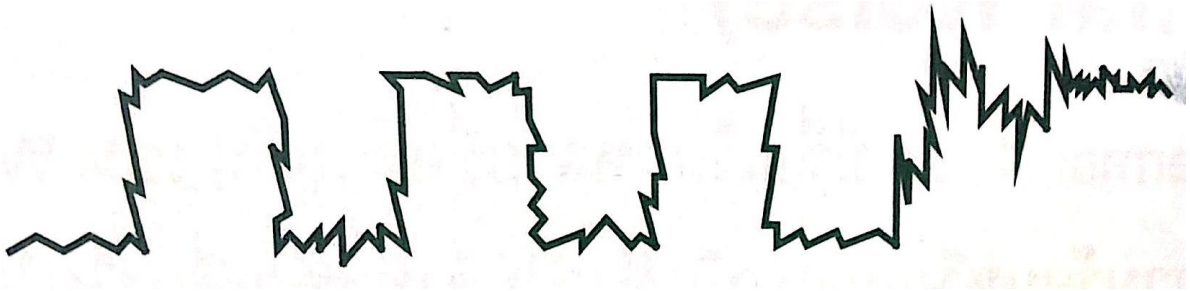
ບົດທີ 4 ການກວດຂໍ້ຜິດພາດຂອງຂໍ້ມູນ ແລະ ເຕັກນິກການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນ (Error detection and flow control)

ສັນຍານລົບກວນເປັນສັນຍານທີ່ມາພ້ອມກັບສັນຍານຂໍ້ມູນເປັນສັນຍານທີ່ບໍ່ພຶງປະສົງເກີດຂຶ້ນໄດ້ຈາກຫຼາຍສາເຫດດ້ວຍກັນເຊັ່ນ: ຈາກສະພາບແວດລ້ອມທີ່ສົ່ງຜົນຕໍ່ຄວາມສ່ຽງໃຫ້ເກີດສັນຍານລົບກວນທາງໄຟ້າ ຫຼື ອຸນຫະພູມ ໂດຍຊະນິດຂອງສັນຍານລົບກວນປະກອບດ້ວຍຊະນິດຕ່າງໆດັ່ງນີ້

1. ສັນຍານລົບກວນ (Noise)

1.1 ເທີ່ມ໋ນອຍສ໌ (Thermal Noise)

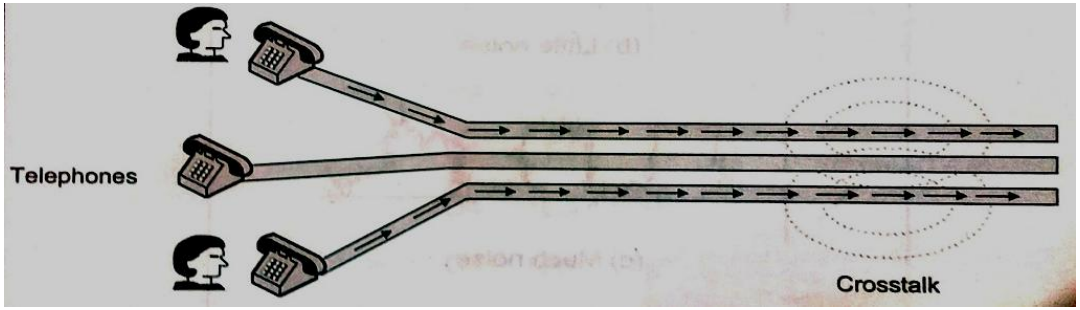
ສັນຍານລົບກວນ Thermal Noise ມີຊື່ອື່ນຫຼາຍດ້ວຍກັນເຊັ່ນ Write Noise ຫຼື Gaussian Noise ເປັນສັນຍານລົບກວນທີ່ເກີດຈາກຄວາມຮ້ອນ ຫຼື ອຸນຫະພູມເຊິ່ງເປັນສິ່ງທີ່ຫຼີກລ້ຽງບໍ່ໄດ້ເລີຍເນື່ອງຈາກເປັນຜົນມາຈາກການເຄື່ອນທີ່ຂອງເອເລັກຕອນໃນເສັ້ນລວດຕ໋ວນ໋ຳ ອຸນຫະພູມສູງລະດັບສັນຍານລົບກວນກໍ່ຈະສູງຂຶ້ນຕາມ ສາມາດປ້ອງກັນໄດ້ດ້ວຍການໃຊ້ອຸປະກອນກອງສັນຍານ (Filters) ສຳລັບສັນຍານອະນາລັອກ ແລະ (Regenerate) ສຳລັບສັນຍານດິຈິຕອນ

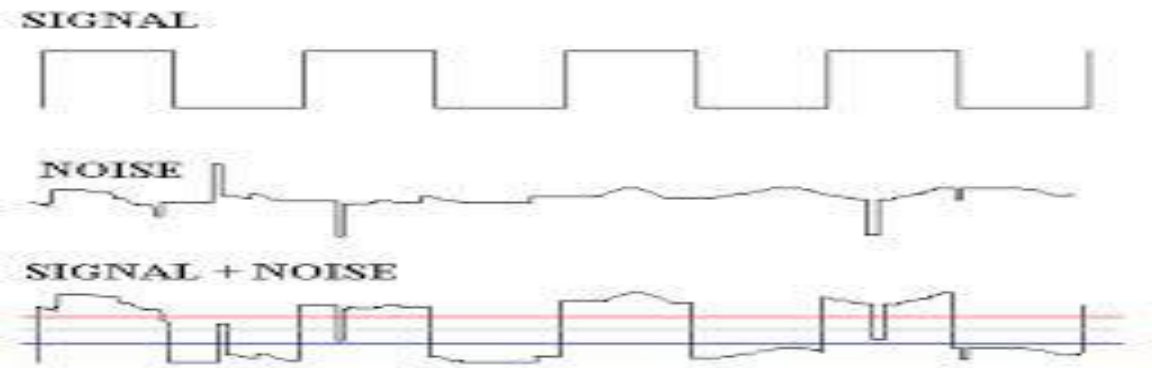


Impulse Noise

1.3 ຄຣອສທ້ອກ (Crosstalk)

ເກີດຈາກການໜ່ວງນຳຂອງສະໜາມແມ່ເຫຼັກໄຟ້າທີ່ເຂົ້າໄປລົບກວນສັນຍານຂໍ້ມູນທີ່ສົ່ງຜ່ານເຂົ້າໄປໃນສາຍສົ່ງເຊັ່ນ ສາຍຄູ່ບິດກຽວ ເຮັດໃຫ້ເກີດການໜ່ວງນຳທາງໄຟ້າເນື່ອງຈາກໃນລະບົບສົ່ງສັນຍານທີ່ມີສາຍສົ່ງຫຼາຍເສັ້ນ. ແຕ່ເຮົາສະມາດປ້ອງກັນໄດ້ດ້ວຍການໃຊ້ສາຍສັນຍານທີ່ມີສະນວນ ຫຼື ມີຊີຣດ໌ (Shield) ເພື່ອປ້ອງກັນສັນຍານລົບກວນ





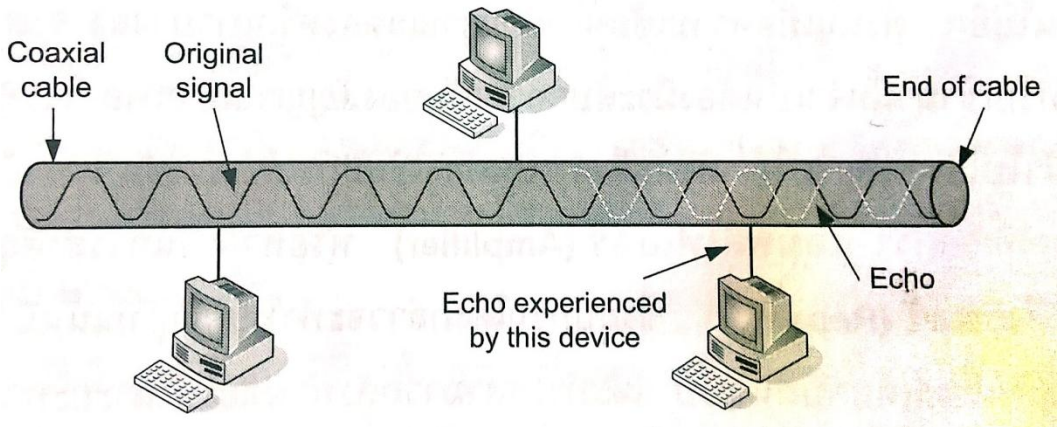
Thermal Noise

1.2 ອິມພັນສ໌ນອຍສ໌ *(Impulse Noise)*

ເປັນສັນຍານທີ່ເກີດຂຶ້ນຢ່າງໄວວາໃນເວລາສັ້ນໆໃດໜຶ່ງແລ້ວກໍ່ຫາຍໄປຈັດເປັນສັນຍານລົບກວນແບບບໍ່ຄົງທີ່ຊຶ່ງຍາກຕໍ່ການກວດສອບ ມັກຈະເກີດການລົບກວນຈາກພາຍນອກເຊັ່ນ: ຟ້າແລບ, ຟ້າຜ່າ. ເຊິ່ງສາມາດປ້ອງກັນໄດ້ດ້ວຍການໃຊ້ອຸປະກອນກອງສັນຍານພິເສດສຳລັບສັນຍານອະນາລັອກ ແລະ ອຸປະກອນປະມວນຜົນສັນຍານສຳລັບສັນນຍານດິຈິຕອນ.

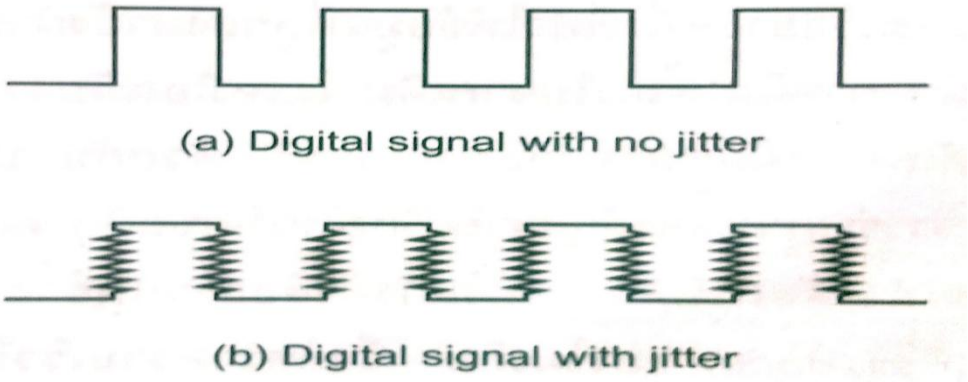
1.4 ເອໂຄ (Echo)

ເປັນສັນຍານທີ່ຖືກສະທ້ອນກັບ (reflection) ສົ່ງຜົນໃຫ້ສັນຍານທີ່ສົ່ງໄປຍ້ອນກັບມາໃນຮູບແບບທີ່ໃຫ້ຄຸ່ງ ເຮົາສາມາດປ້ອງກັນໄດ້ດ້ວຍການໃຊ້ອຸປະກອນເທີມີເນດຕີ (Termainator) ເຊັ່ນໃນລະບົບເຄືອຂ່າຍທ້ອງຖິ່ນທີ່ມີການໃຊ້ສາຍໂຄເອັກຊຽວເປັນຕົວກາງສົ່ງຂໍ້ມູນຈະໃຊ້ເທີມີເນເຕີທີ່ປາຍສາຍທັງສອງເພື່ອລະງັບສຽງສະທ້ອນດັ່ງກ່າວດ້ວຍການດູດກືນສັນຍານເຫຼົ່ານັ້ນ.



1.5 ຈິດເຕີ (Jitter)

ເປັນເຫດການທີ່ຄວາມຖີ່ຂອງສັນຍານໄດ້ມີການປ່ຽນແປງຢ່າງຕໍ່ເນື່ອງເຊິ່ງ ເຮັດໃຫ້ເກີດການເລື່ອນເຟສເປັນຄ່າອື່ນ ສຳລັບການປ້ອງກັນສັນຍານລົບກວນນີ້ ສາມາດປ້ອງກັນໄດ້ດ້ວຍການເລືອກໃຊ້ວົງຈອນເອເລັກໂຕນິກທີ່ມີຄຸນນະພາບ ຫຼື ອາດຈະໃຊ້ອຸປະກອນລີພີສເຕີ.



1.6 ຄວາມຜຽນຈາກການເຄື່ອນທີ່ (Delay Distortion)

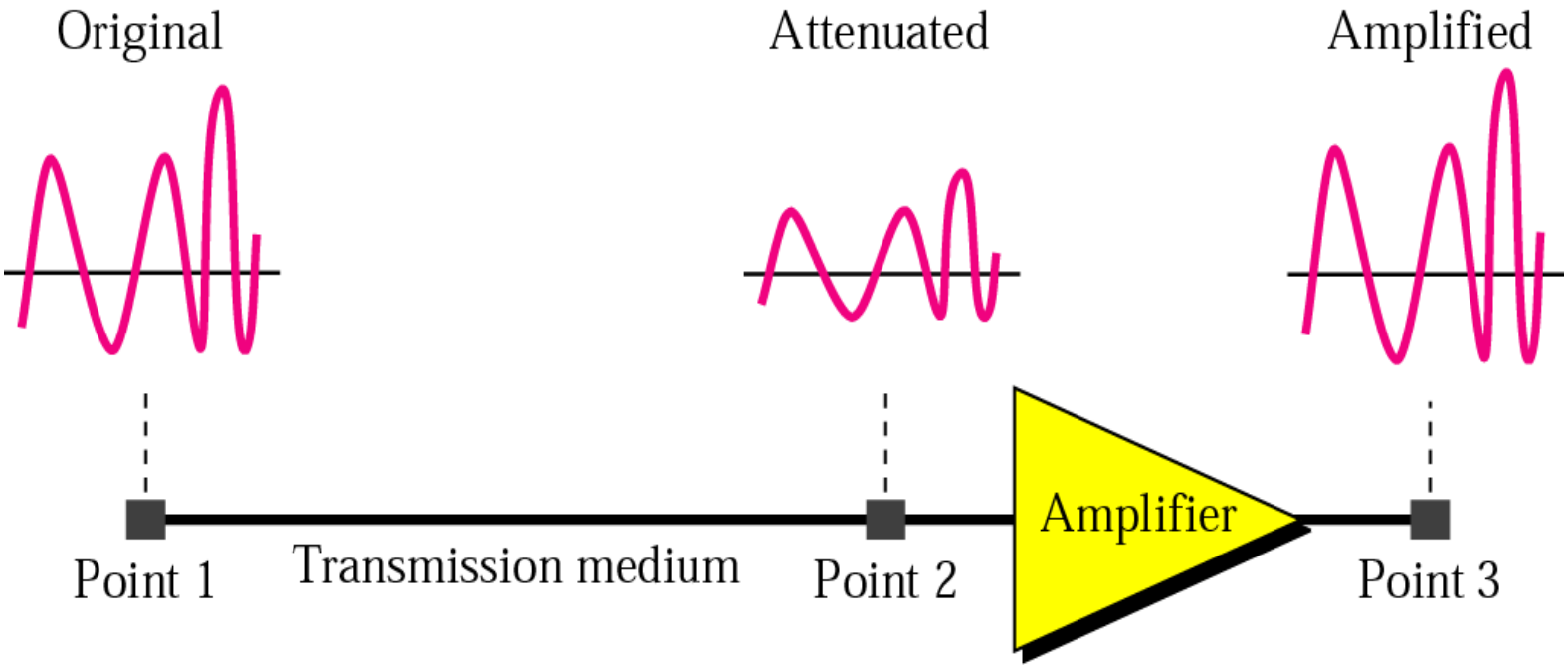
ເປັນເຫດການໜຶ່ງທີ່ສາມາດເກີດຂຶ້ນໄດ້ເນື່ອງມາຈາກສັນຍານຂໍ້ມູນເຄື່ອນທີ່ດ້ວຍຄວາມໄວແຕກຕ່າງກັນ ສົ່ງຜົນຕໍ່ຄວາມຜິດພາດຂອງຂໍ້ມູນ ສຳລັບການປ້ອງກັນສັນຍານລົບກວນຊະນິດນີ້ສາມາດປ້ອງກັນໄດ້ດ້ວຍການເພີ່ມວົງຈອນ Equalizer ເພື່ອກວດສອບສັນຍານທີ່ເຂົ້າມາ ແລະ ທຳການປັບຄວາມໄວຂອງຄວາມຖີ່ໃຫ້ເທົ່າກັນ.

1.7 ການອ່ອນກຳລັງຂອງສັນຍານ (Attenuation)

ການເດີນທາງຂອງສັນຍານໃນໄລຍະທາງໄກໆເຮັດໃຫ້ເກີດການອ່ອນກຳລັງຂອງສັນຍານ ຈະເຮັດໃຫ້ຄວາມເຂັ້ມຂອງສັນຍານລົດລົງສົ່ງຜົນກະທົບຕໍ່ອຸປະກອນຮັບ ແລະຜູ້ຮັບ ດັ່ງນັ້ນ ຈຶ່ງຈຳເປັນຕ້ອງໃຊ້ອຸປະກອນຊ່ວຍເຊັ່ນ : ຫາກເປັນການສົ່ງສັນຍານແບບອະນາລັອກຈະໃຊ້ອຸປະກອນ ແອມພິໄຟເອີ (Amplifier) ຫຼື ຫາກເປັນສັນຍານດິຈິຕອນກໍ່ຈະໃຊ້ອຸປະກອນ ຣີພີເຕີ (Repeater) ເຊິ່ງອຸປະກອນດັ່ງກ່າວຈະເຮັດໃຫ້ສັນຍານນັ້ນມີກຳລັງສົ່ງເຮັດໃຫ້ສາມາດສົ່ງຕໍ່ໄປໃນໄລຍະທາງໄກຕໍ່ໄປໄດ້ອີກຕາມໄລຍະທາງທີ່ກຳນົດ.

Attenuation

AMPLIFIER


Original
Attenuated
Amplified
Amplifier
Point 1
Transmission medium
Point 2
Point 3

2. ເທັກນິກການກວດຈັບຂໍ້ຜິດພາດ (Error Detection Techniques)

ໃນການກວດຈັບຂໍ້ຜິດພາດນີ້ແມ່ນຈະໃຊ້ລະຫັດສຳລັບກວດສອບຂໍ້ຜິດພາດເຊິ່ງມີຫຼາຍວິທີດັ່ງນີ້:

2.1 ການໃຊ້ບິດກວດສອບ (Parity Check)

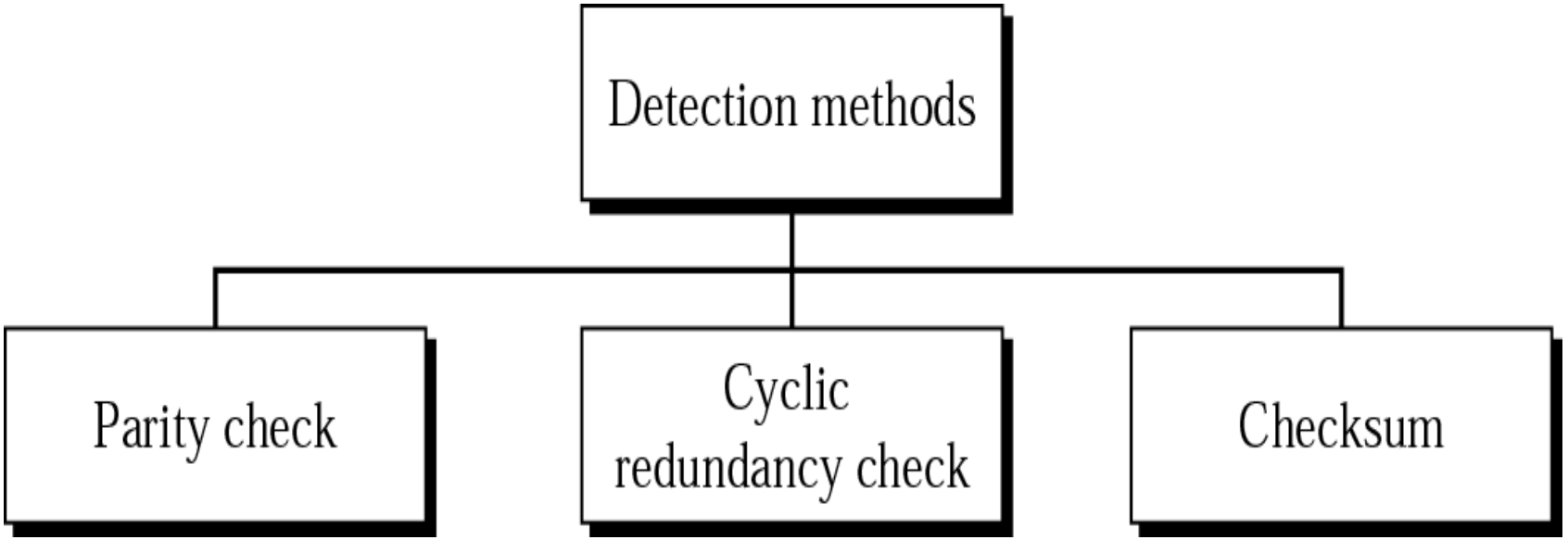
ເປັນວິທີ ຫຼືເທັກນິກການກວດຈັບທີ່ງ່າຍ ແລະເປັນວິທີທີ່ເກົ່າແກ່ໂດຍຈະໃຊ້ບິດພາລິຕີ ປະກອບດ້ວຍ 0 ແລະ 1 ປະທ້າຍເພີ່ມເຂົ້າມາອີກໜຶ່ງບິດເພື່ອໃຊ້ເປັນບິດກວດສອບໂດຍມີວິທີກວດສອບຢູ່ສອງວິທີ

+ **ການກວດສອບບິດພາວະຄູ່** (Even Parity)

+ **ການກວດສອບບິດພາວະຄີກ** (Odd Parity)

ຕົວຢ່າງເຊັ່ນຖ້າຂໍ້ມູນເທົ່າກັບ 0100110 ໂດຍສົມມຸດການກວດສອບບິດແບບພາວະຄູ່ ດັ່ງນັ້ນບິດທີ່ຈະເພີ່ມເຂົ້າໄປຄື 1 ຈະໄດ້ 01001101. ແຕ່ຖ້າຂໍ້ມູນມີຄ່າເທົ່າກັບ 0110110 ບິດພາຣິຕີເພີ່ມເຂົ້າໄປກໍ່ຄື 0 ເນື່ອງຈາກຜົນລວມຂອງບິດ 1 ເປັນ 4 ເຊິ່ງເປັນເລກຄູ່ຢູ່ແລ້ວ ຈະໄດ້ 01101100 ແຕ່ສຳລັບການໃຊ້ບິດກວດສອບດ້ວຍວິທີການກວດສອບບິດແບບພາວະຄີກກະຈະມີລັກສະນະຄ້ຽວກັນ ແຕ່ເປັນໄປໃນທາງກົງກັນຂ້າມ.

Detection methods



Original Data	Sender parity bit	Transmitted Information	Receiver calculated parity bit	Agree?
0100110	1	01001101	1	Yes
0100110	1	01001001	0	No

ສະແດງການກວດສອບບິດພາວະຄູ່ເຊິ່ງມີບິດໜຶ່ງເກີດການປ່ຽນແປງ

ຢ່າງໃດກໍ່ຕາມ ການກວດສອບນີ້ກໍ່ມີຂໍ້ເສຍເຊັ່ນກັນ ຫາກມີຂໍ້ມູນເກີດຜິດພາດຈຳນວນສອງບິດ ຫຼື ເກີດຂໍ້ຜິດພາດຫຼາຍໆ ຈະເຮັດໃຫ້ບໍ່ສາມາດກວດພົບຂໍ້ຜິດພາດ. ດັ່ງຕົວຢ່າງ

Original Data	Sender parity bit	Transmitted Information	Receiver calculated parity bit	Agree?
0100110	1	01001101	1	Yes
0100110	1	01000001	0	Yes

2.2 ການຫາຜົນລວມ (Checksum)

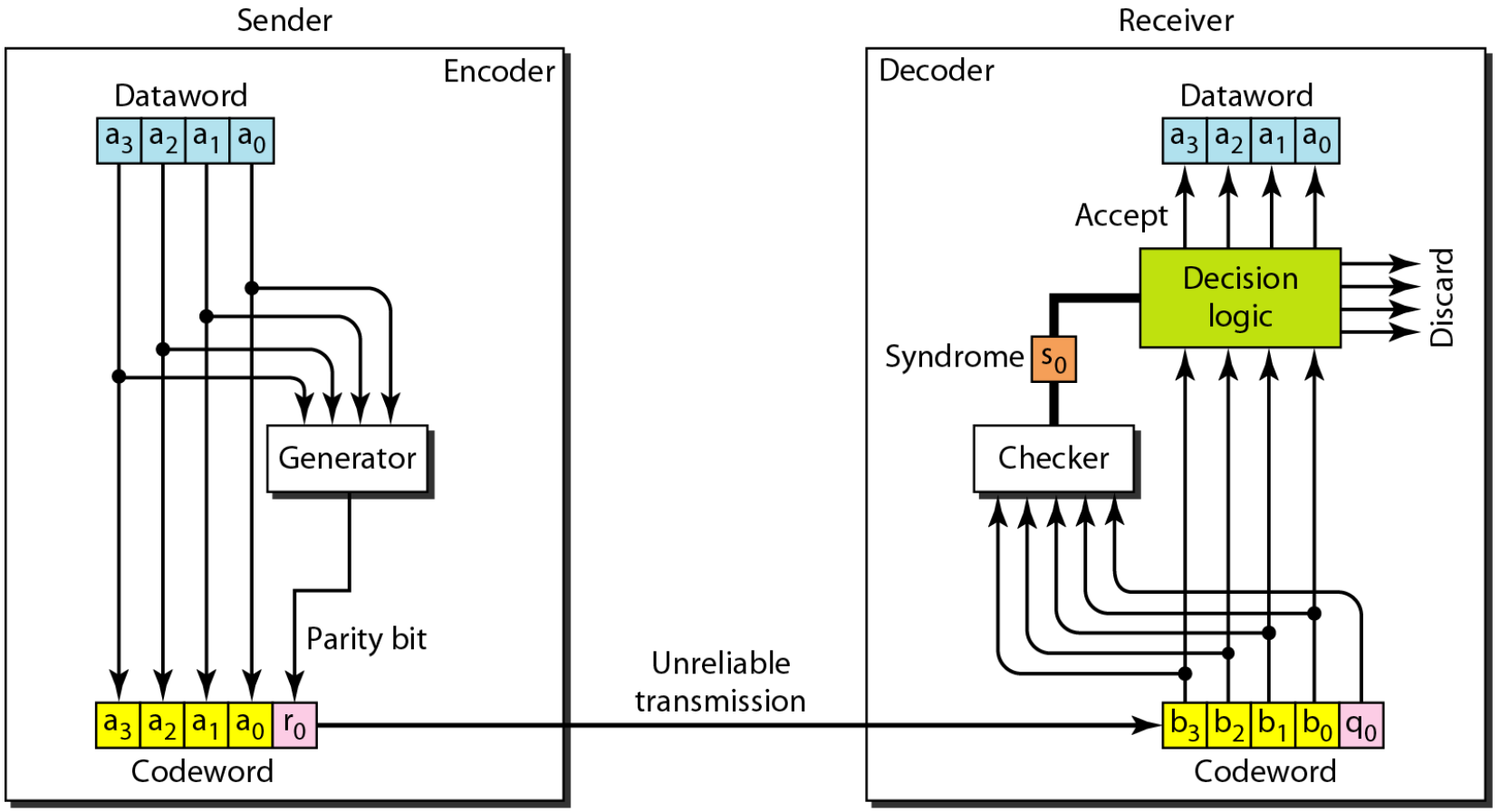
ວິທີການຫາຜົນລວມນີ້ເປັນວິທີໜຶ່ງທີ່ມີປະສິດທິພາບສູງກວ່າວິທີການໃຊ້ບິດກວດສອບ ຢ່າງໃດກໍ່ຕາມວິທີນີ້ກໍ່ມີຂໍ້ເສຍເຖິງແມ້ວ່າຜົນລວມທີ່ຄຳນວນໄດ້ນັ້ນມີຄ່າຄືກັນກໍ່ຈິງຢູ່ ແຕ່ຖ້າຫາກຄ່າຂອງຂໍ້ມູນແຕ່ລະຕົວໄດ້ມີການປ່ຽນແປງ ແລະ ເກີດຜົນລວມເທົ່າກັນ ກໍ່ຈະກວດບໍ່ພົບຂໍ້ຜິດພາດດັ່ງຕົວຢ່າງ

Data item in binary	Checksum value		Data item in binary	Checksum value
0101	5		0100	4
0110	6		0111	7
0100	4		0101	5
0001	1		0000	0
Total	16		Total	16

A B

ສະແດງການກວດຈັບຂໍ້ຜິດພາດດ້ວຍ Checksum ແຕ່ກວດຈັບຂໍ້ຜິດພາດບໍ່ເຫັນເນື່ອງຈາກຂໍ້ມູນເກີດການປ່ຽນແປງ

ENCODER AND DECODER FOR SIMPLE PARITY CHECK CODE



2.3 ການໃຊ້ວິທີ CRC (Cyclic Redundancy Checksum)

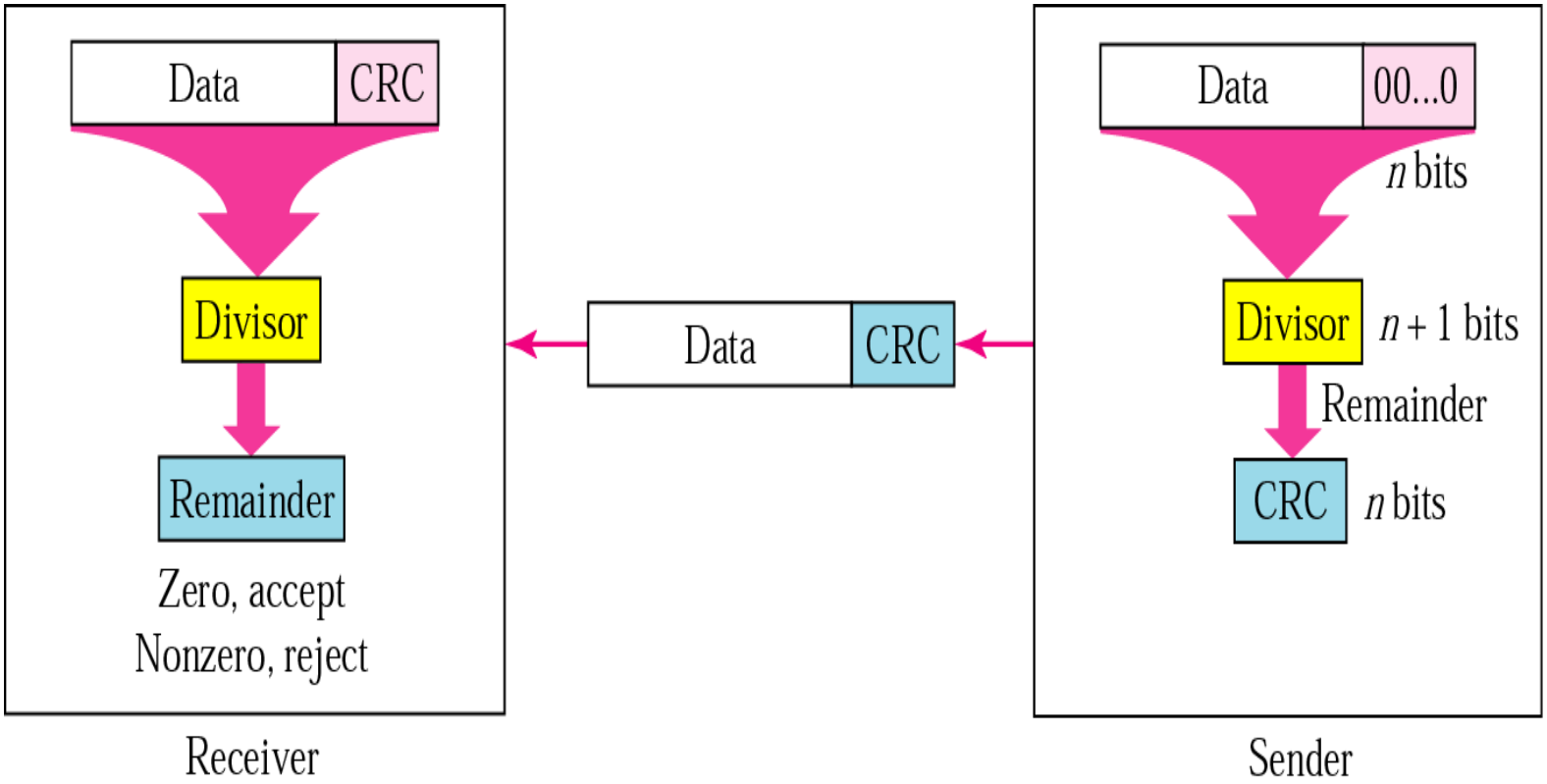
ຈາກຫຼາຍວິທີທີ່ໄດ້ກ່າວມາຈະມີຂໍ້ເສຍທີ່ບໍ່ສາມາດກວດຈັບຂໍ້ຜິດພາດໄດ້ເລີຍໃນກໍລະນີທີ່ເກີດບິດຜິດພາດແບບສອງບິດຂຶ້ນໄປຫຼືເອີ້ນວ່າ: Burst Error ເຮັດໃຫ້ປະສິດທິພາບໃນການກວດຈັບຕ່ຳລົງ ດັ່ງນັ້ນ ຈຶ່ງເກີດມີວິທີການກວດຈັບຂໍ້ຜິດພາດແບບ CRC ເຊິ່ງເປັນວິທີໜຶ່ງທີ່ນິຍົມໃຊ້ໃນເຄືອຂ່າຍທ້ອງຖິ່ນ ແລະ ຈັດເປັນວິທີທີ່ມີປະສິດທິພາບສູງກວ່າວິທີການໃຊ້ລະຫັດກວດສອບ ແລະ ການຫາຜົນລວມ. ໂດຍສະເພາະ CRC-32 ບິດ ນັ້ນມີອັດຕາຄວາມແນ່ນອນໃນການກວດຈັບຂໍ້ຜິດພາດໄດ້ຫຼາຍເຖິງ 99.99999998% ແລະ ນິຍົມໃຊ້ໃນເຄືອຂ່າຍອິນເຕີເນັດ.

ຮູບແບບຂອງເຟຣມຂໍ້ມູນທີ່ມີການໃຊ້ລະຫັດກວດຈັບຂໍ້ຜິດພາດ

Soh	data	eot	CRC

CRC: Cyclic Redundancy Check

CRC = Remainder



ຫຼັກການຂອງ CRC ຈະໃຊ້ລະຫັດໂພລີໂນເມຍ (Polynomial Codes) ເຊິ່ງຕ້ອງມີຄຸນສົມບັດໂດຍບິດຊ້າຍສຸດ ແລະບິດຂວາສຸດຕ້ອງມີຄ່າເປັນ 1 ສະເໝີ ແລະ ລະຫັດໂພລີໂນເມຍຈະຕ້ອງມີຈຳນວນບິດທີ່ນ້ອຍກວ່າຈຳນວນບິດຂອງຂໍ້ມູນ ໃນການຄຳນວນໂພລີໂນເມຍຈະເປັນລັກສະນະ Exclusive-OR ໂດຍຈະບໍ່ມີການໃຊ້ Carry Bit ທັງການບວກ ແລະການລົບ. ໂດຍບິດທີ່ນຳມາບວກຫຼືລົບກັນ ຫາກບິດຄືກັນຜົນທີ່ໄດ້ເທົ່າກັບ 0 ຫາກບິດຕ່າງກັນຜົນທີ່ໄດ້ຈະມີຄ່າເທົ່າກັບ 1.

ຜົນການປະຕິບັດຂອງການ Exclusive-OR ຂອງຕົວເລກສອງຄ່າ

First input	Second input	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0

ວິທີການຄຳນວນຫາ CRC

ຈະນຳສົ່ງຜົນໂພລີໂນເມຍທີ່ກຳນົດຂຶ້ນ G(x) ໄປຫານກັບເຟຣມຂໍ້ມູນທີ່ຕ້ອງການສົ່ງ M(x) ທີ່ລວມກັບບິດສູນ (n) ເພີ່ມເຕີມໄວ້ແລ້ວ, ໂດຍຜົນທີ່ໄດ້ຮັບຈະຖືກຖິ້ມໄປໃຫ້ພິຈາລະນາພຽງເລກເສດທີ່ໄດ້ຈາກການຫານ R(x) ຫຼື Remainder ເທົ່ານັ້ນ. ໂດຍເຟຣມຂໍ້ມູນທີ່ສົ່ງໄປຍັງປາຍທາງເອີ້ນວ່າ T(x) ທີ່ໄດ້ຈາກການນຳ M(x) ແລະ ເພີ່ມຕໍ່ທ້າຍດ້ວຍເຟຣມ R(x) ເອີ້ນວ່າ: FCS (Frame Check Sequence) ເຊິ່ງໂດຍທົ່ວໄປເອີ້ນວ່າ: CRC ນັ້ນເອງ.

ເມື່ອສະຖານີຕົ້ນທາງໄດ້ທຳການສົ່ງເຟຣມ T(x) ຜ່ານການກວດສອບຂໍ້ຜິດພາດແບບ CRC ໄປຍັງສະຖານີປາຍທາງ ຝ່າຍສະຖານີປາຍທາງກໍ່ຈະໃຊ້ລະຫັດໂພລີໂນເມຍທີ່ເປັນລະຫັດຊະນິດດຽວກັນກັບຝ່າຍສົ່ງ ໄປທຳການກວດສອບຂໍ້ຜິດພາດດ້ວຍການຄຳນວນ ໂດຍການນຳ T(x) ຫານດ້ວຍ G(x) ເຊິ່ງຈະພິຈາລະນາຈາກເສດທີ່ໄດ້ຈາກການຫານ ໂດຍຫາກເສດທີ່ໄດ້ມີຄ່າເປັນສູນ ນັ້ນໝາຍເຖິງຂໍ້ມູນທີ່ໄດ້ຮັບນັ້ນຖືກຕ້ອງ

M(x) ຄືເຟຣມຂໍ້ມູນທີ່ຕ້ອງການສົ່ງ (Message to be Transmitted)

G(x) ຄືໂພລີໂນເມຍທີ່ຕັ້ງຂຶ້ນມາ (Generator Polynomial)

n ຄືບິດສູນທີ່ເພີ່ມເຕີມດ້ວຍການນຳໄປປະທ້າຍເຟຣມ M(x) ໂດຍຈຳນວນບິດພິຈາລະນາຈາກເລກກຳລັງ (Degree) ຂອງ G(x)

R(x) ຄືຜົນເສດທີ່ໄດ້ຈາກການຄຳນວນ (Remainder) (ໄດ້ຈາກການນຳ M(x) ທີ່ໄດ້ລວມກັບບິດສູນ (n) ເພີ່ມເຕີມ ແລ້ວຫານດ້ວຍ G(x))

T(x) ຄືເຟຣມທີ່ສົ່ງໄປ (Transmitted Frame) ເຊິ່ງໄດ້ຈາກການນຳ M(x) ປະທ້າຍດ້ວຍ R(x) ຫຼື T(x)=M(x)+R(x)

ຈາກຕົວຢ່າງນີ້ ສົມມຸດວ່າ ເຟຣມຂໍ້ມູນທີ່ຕ້ອງການສົ່ງໄປຄື 1101011011 ແລະໃຊ້ໂພລີໂນເມຍ X^4+X+1 ດັ່ງນັ້ນ

M(x) = 1101011011

G(x) = X^4+X+1

= $(1 \times X^4)+(0 \times X^3)+(0 \times X^2)+(1 \times X^1)+(1 \times X^0)$

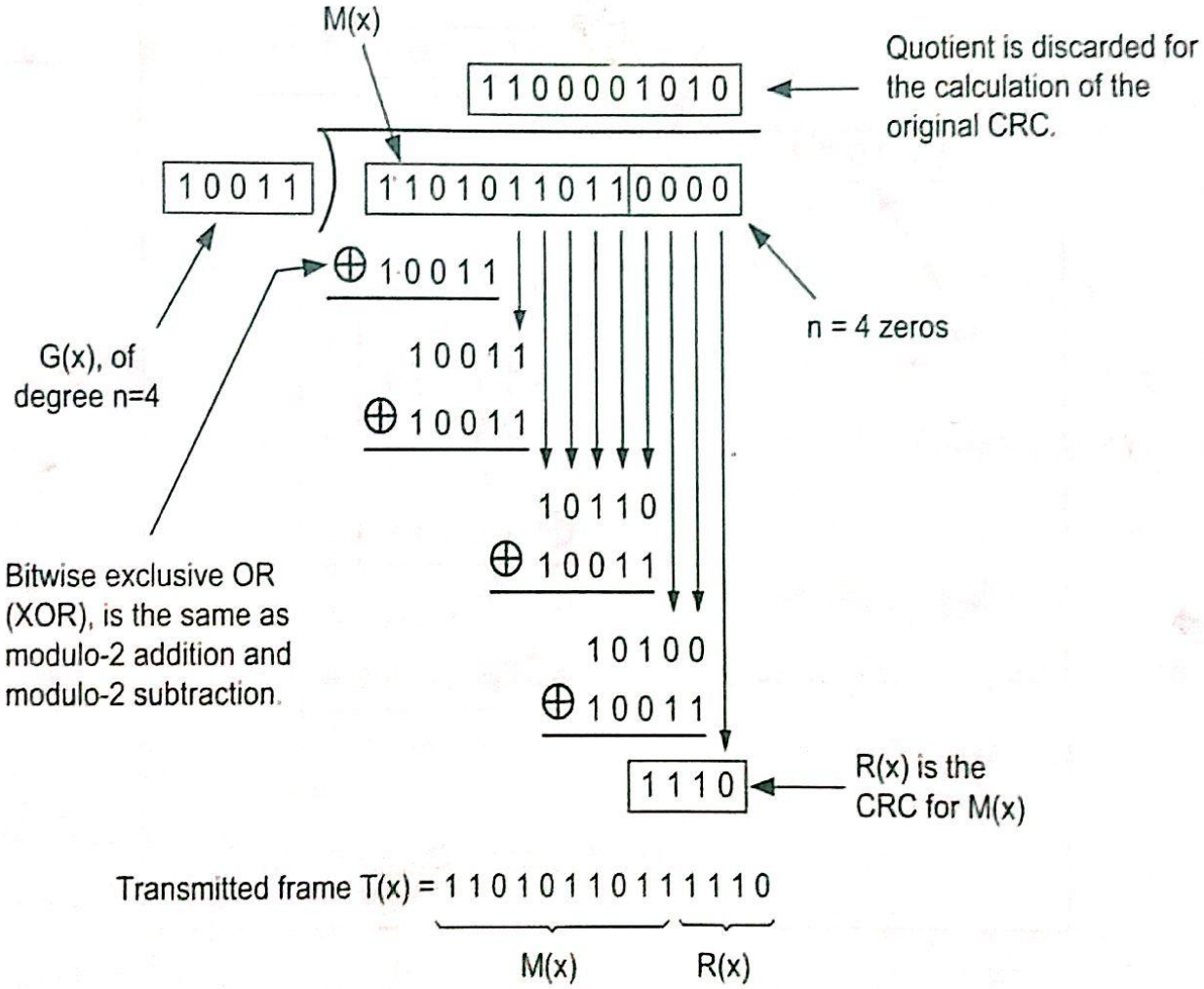
=10011

ເມື່ອພິຈາລະນາຄຸນສົມບັດກໍ່ສາມາດສະແດງລາຍລະອຽດໄດ້ດັ່ງຕໍ່ໄປນີ້:

ບິດຊ້າຍສຸດແລະບິດຂວາສຸດຂອງ G(x) ມີຄ່າເປັນ 1 ນັ້ນຖືວ່າຕົງຕາມຄຸນສົມບັດ

ຈຳນວນບິດຂອງ G(x) ມີນ້ອຍກວ່າ M(x) ນັ້ນຖືວ່າຕົງຕາມຄຸນສົມບັດກຳລັງຂອງ G(x) ເທົ່າກັບ 4 ດັ່ງນັ້ນ n ຈຶ່ງມີຄ່າເທົ່າກັບ 4 ໂດຍໃຫ້ເພີ່ມບິດສູນຈຳນວນ 4 ບິດຕໍ່ທ້າຍເຟຣມ M(x) ຈະໄດ້ 1101011011 0000

ເມື່ອທຳການພິຈາລະນາຄຸນສົມບັດ ແລະຫາຄ່າຕ່າງໆ ດັ່ງເບື້ອງຕົ້ນແລ້ວກໍ່ດຳເນີນການຫາ CRC ດັ່ງລຸ່ມນີ້:



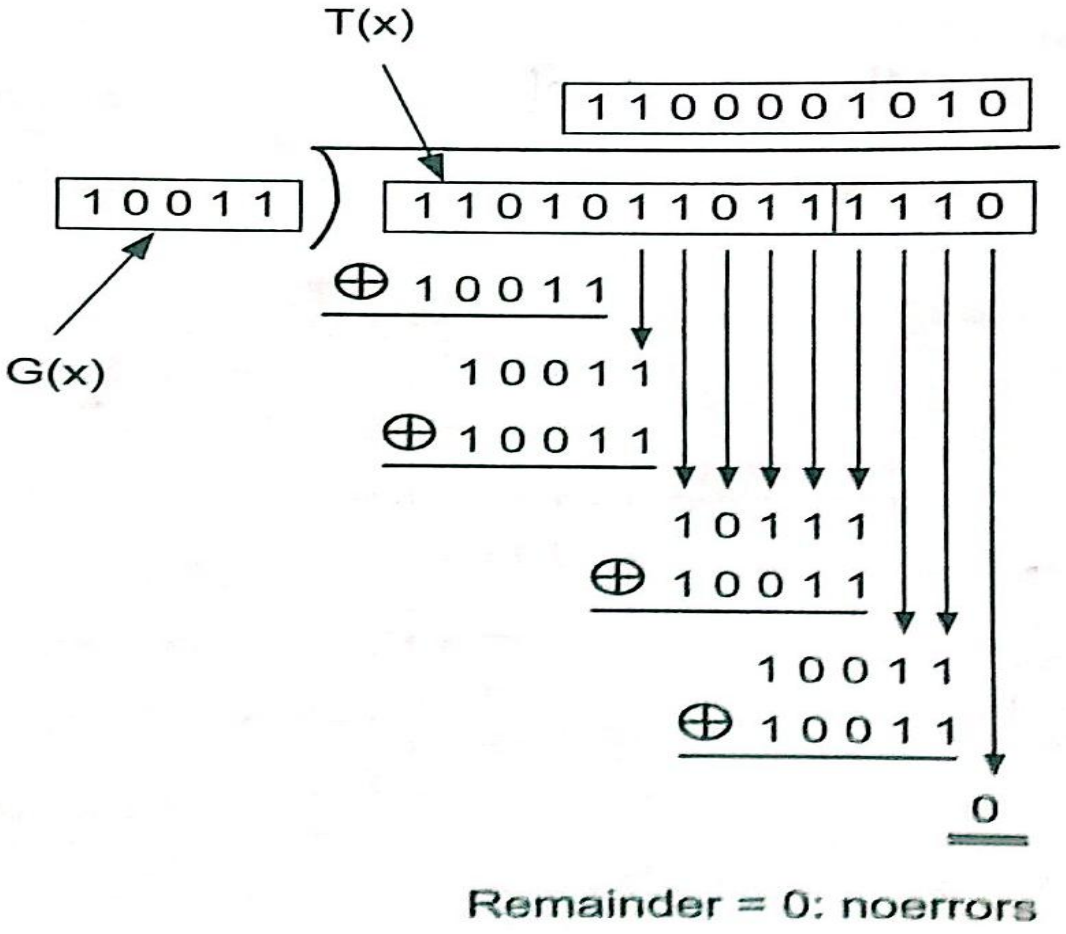
ຈາກຄ່າທີ່ຄຳນວນໄດ້ ດັ່ງນັ້ນ ເຟຣມທີ່ຈະສົ່ງໄປຍັງປາຍທາງຄື:

$$T(x) = M(x) + R(x)$$

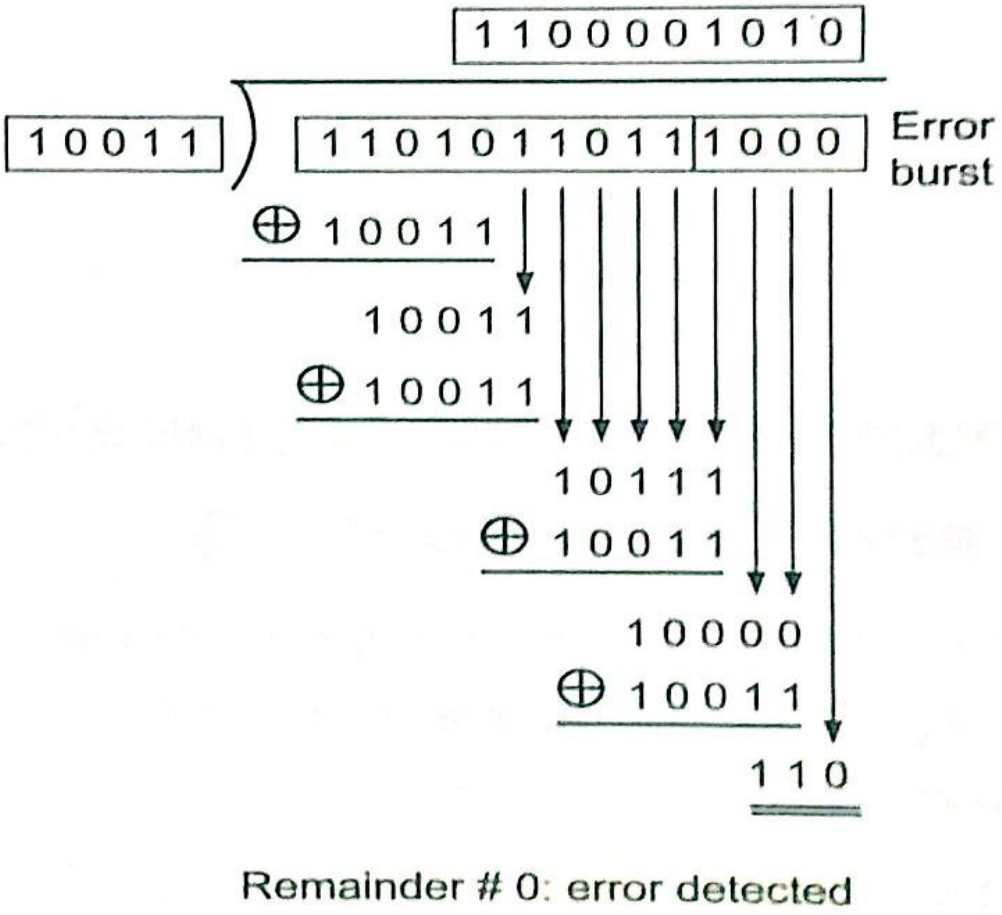
$$= 1101011011\ 1110$$

ຫຼັງຈາກທີ່ໄດ້ທຳການຄຳນວນຫາ CRC ເປັນທີ່ຮຽບຮ້ອຍ ຝ່າຍສົ່ງກໍ່ຈະສົ່ງ T(x) ໄປຍັງປາຍທາງ ເມື່ອປາຍທາງໄດ້ຮັບເຟຣມດັ່ງກ່າວ ກໍ່ຈະນຳ T(x) ໄປຫານດ້ວຍໂພລີໂນເມຍ ຫຼືG(x) ທີ່ເປັນລະຫັດດຽວກັນກັບຝ່າຍສົ່ງໂດຍຜົນລັບຈາກການຄຳນວນ T(x)/G(x) ຈະຕ້ອງຫານລົງຕົວ ຫຼື ມີເສດເປັນສູນແມ່ນໝາຍຄວາມວ່າບໍ່ມີການຜິດພາດຂອງບິດຂໍ້ມູນ. ແຕ່ຖ້າຫາກວ່າການຄຳນວນບໍ່ລົງຕົວເສດທີ່ໄດ້ບໍ່ມີຄ່າເທົ່າສູນນັ້ນສະແດງວ່າມີການຜິດພາດຂອງບິດຂໍ້ມູນເກີດຂຶ້ນ.

ฝ่ายสะถานิธับจะทำหานหาดสอบถามถืกต้องด้วยหานนำ T(x)/G(x) โดยบ่พิบข้ผิดพาดเมื่องจาหเสดมีถ่าเป้นสูน



ສະແດງການກວດຈັບຂໍ້ຜິດພາດເນື່ອງຈາກຜົນຈາກການຄຳນວນບໍ່ລົງຕົວ



Remainder # 0: error detected

ຢ່າງໃດກໍ່ຕາມຈາກຕົວຢ່າງເປັນຕົວຢ່າງທີ່ໃຊ້ລະຫັດໂພລີໂນເມຍທີ່ມີດີກຼີເທົ່າກັບ 4 ($G(x)= X^4+X+1$) ເພື່ອໃຫ້ງ່າຍຕໍ່ການຄຳນວນ ແລະນຳໄປໃຊ້ປະກອບເປັນຕົວຢ່າງ ແຕ່ມາດຕະຖານໂພລີໂນເມຍທີ່ໃຊ້ງານແບບສາກົນ ແລະສາມາດຮັບປະກັນຂໍ້ຜິດພາດໄດ້ເປັນຢ່າງດີນັ້ນປະກອບດ້ວຍ

- CRC-16 = $X^{16}+X^{15}+X^2+1$

1100000000000101

- CRC-CCITT = $X^{16}+X^{12}+X^5+1$

1000100000100001

- CRC-32 = $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

100000100110000010001110110111

ໂດຍສະເພາະ CRC-32 ນັ້ນສາມາດຮັບປະກັນຄວາມຖືກຕ້ອງໃນການກວດຈັບຂໍ້ຜິດພາດໄດ້ຫຼາຍເຖິງ 99.99% ແລະມີການນຳໄປໃຊ້ງານເທິງເຄືອຂ່າຍທ້ອງຖິ່ນ ຫຼືອີເທີເນັດ

3. ການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນ ແລະ ການຄວບຄຸມຂໍ້ຜິດພາດ (Flow Control and Error Control)

ໃນການສື່ສານລະຫວ່າງກັນໃນເຄືອຂ່າຍຂໍ້ຜິດພາດຕ່າງໆອາດເກີດຂຶ້ນໄດ້ຈາກປັດໃຈຕ່າງໆ ໂດຍສາເຫດທີ່ຕ້ອງມີການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນ ແລະການຄວມຄຸມຂໍ້ຜິດພາດ ກໍ່ເນື່ອງມາຈາກ.

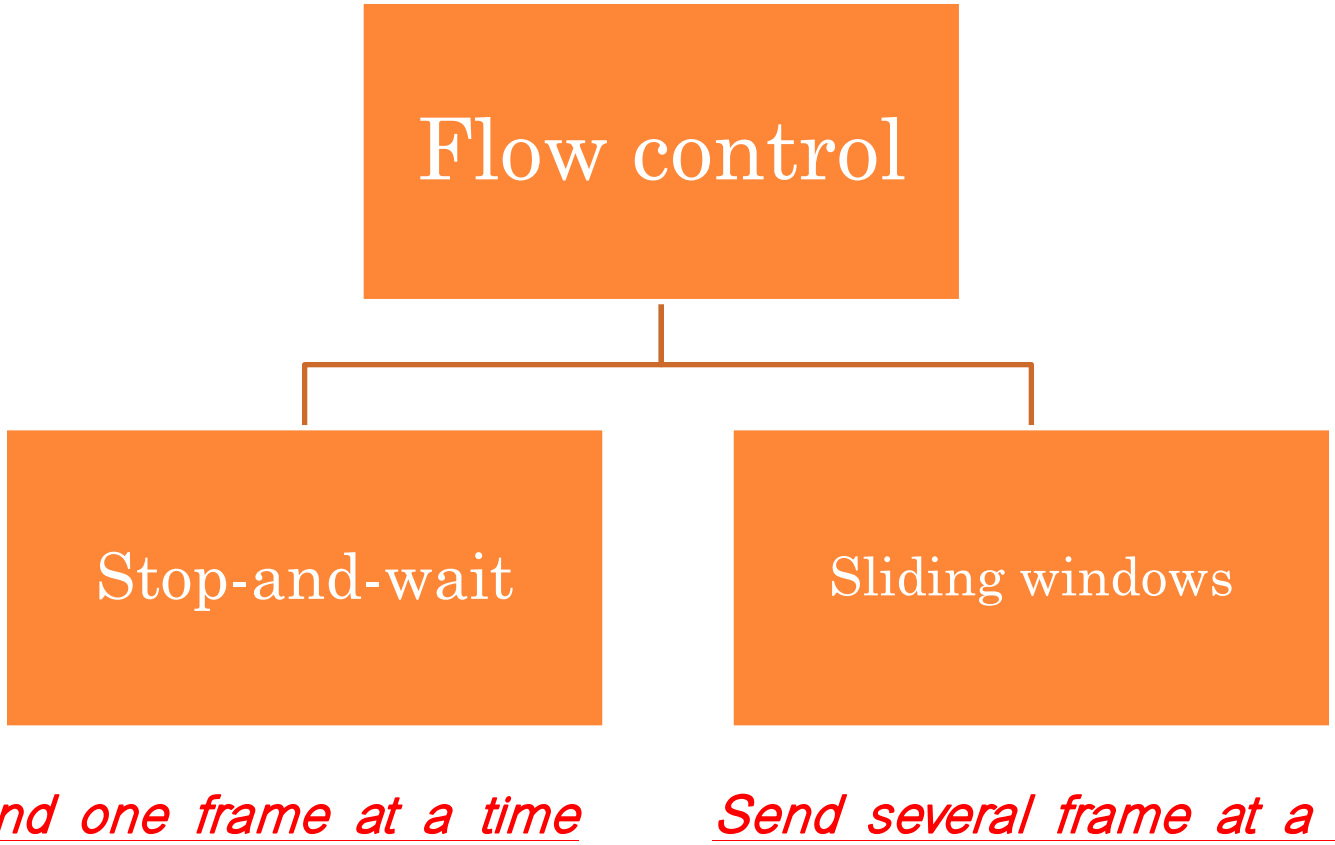
1. ໃນກໍລະນີຝ່າຍສົ່ງ ແລະຝ່າຍຮັບ ສື່ສານຢູ່ໃນຄວາມໄວທີ່ແຕກຕ່າງກັນ
2. ຈະທໍາການໂຕ້ຕອບກັນແນວໃດຫາກເຟຣມຂໍ້ມູນທີ່ສົ່ງໄປນັ້ນເກີດຄວາມເສຍຫາຍ ຫຼື ສູນເສຍ
3. ຈະເກີດຫຍັງຂຶ້ນຫາກຝ່າຍຮັບບໍ່ຮູ້ວ່າມີຂ່າວສານສົ່ງມາເຖິງຕົນເອງ
4. ຈະເກີດຫຍັງຂຶ້ນຖ້າເຟຣມ້ມູນຂອງຝ່າຍສົ່ງນັ້ນເກີດຄວາມເສຍຫາຍ

3.1 ການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນ (Flow Control)

ການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນ ແມ່ນກຸ່ມວິທີຂອງການທີ່ຈະບອກຝ່າຍສົ່ງວ່າ ຈະສາມາດສົ່ງຂໍ້ມູນຈຳນວນເທົ່າໃດກ່ອນທີ່ຈະໄດ້ຮັບການຮັບຮອງ (Acknowledgment) ຈາກຝ່າຍຮັບ ໂດຍການຄວບຄຸມນີ້ຈະຕ້ອງບໍ່ໃຫ້ຝ່າຍຮັບໄດ້ຮັບຂໍ້ມູນຫຼາຍຈົນເກີນ. ດ້ວຍເຫດນີ້ອຸປະກອນຝ່າຍຮັບຈຶ່ງຕ້ອງມີບັອກໜ່ວນຄວາມຈຳ (Buffer) ເພື່ອຈອງໄວ້ສຳລັບເກັບຂໍ້ມູນທີ່ຫຼັ່ງໄຫຼເຂົ້າມາຈົນກະທັ່ງປະມວນຜົນແລ້ວ ແລະຖ້າຫາກໜ່ວຍຄວາມຈຳບັບເຟີເຕັມ ຝ່າຍຮັບຕ້ອງສາມາດບອກຝ່າຍສົ່ງໃຫ້ຫຍຸດການສົ່ງ ໂດຍຈະສະຫຼຸບໄດ້ວ່າ ການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນນີ້ຈະດຳເນີນກ່ຽວກັບ.

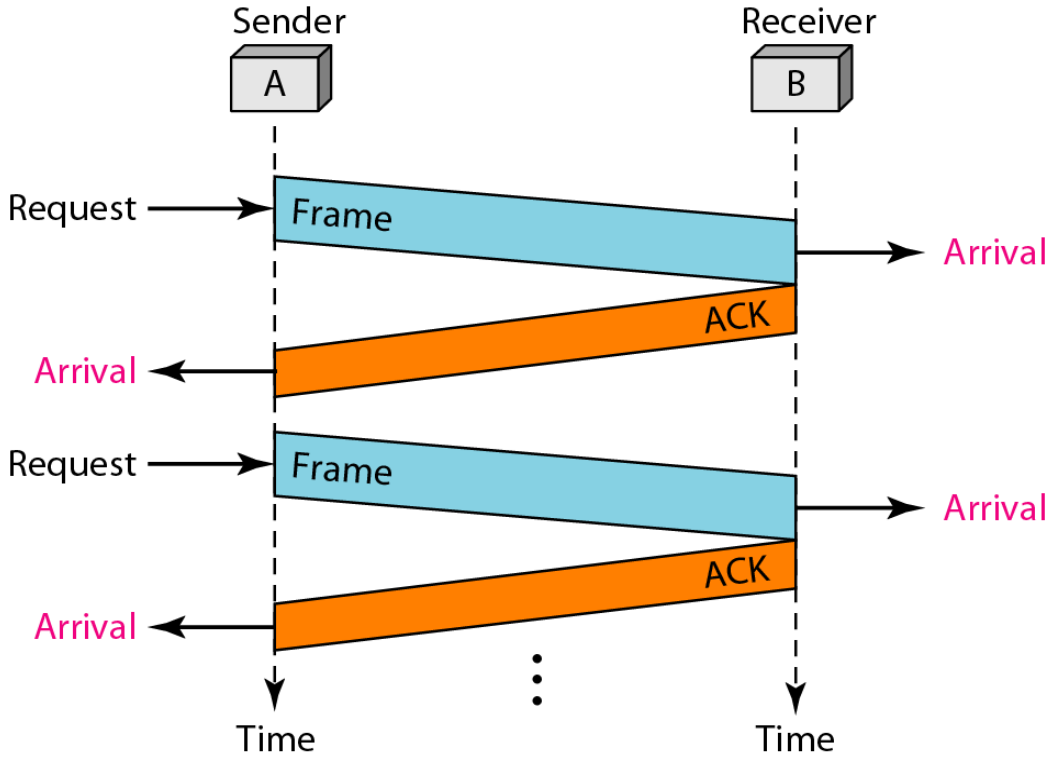
- ກຳໜົດແຜນການສົ່ງເຟຣມຕ່າງໆ ທີ່ຕ້ອງການສົ່ງແລະທຳການຕິດຕາມ
- ສົ່ງເຟຣມເມື່ອໃດ
- ເຟຣມທີ່ສົ່ງຈະຈົບ ຫຼືສິ້ນສຸດເມື່ອໃດ

ປະເພດຂອງການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນ

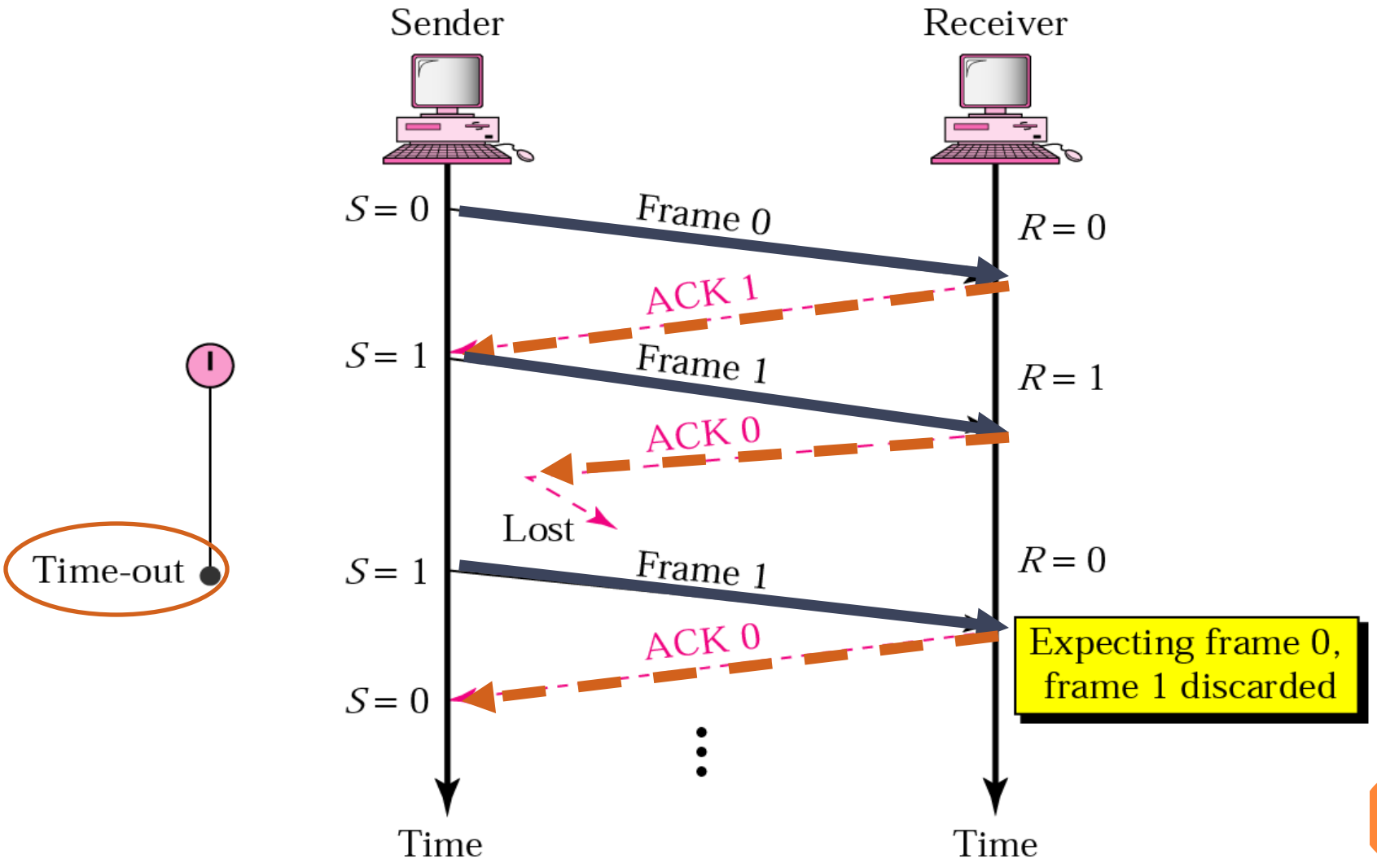


3.1.1 ການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນດ້ວຍວິທີການຫຍຸແລະລໍ (Stop-and-Wait Flow Control)

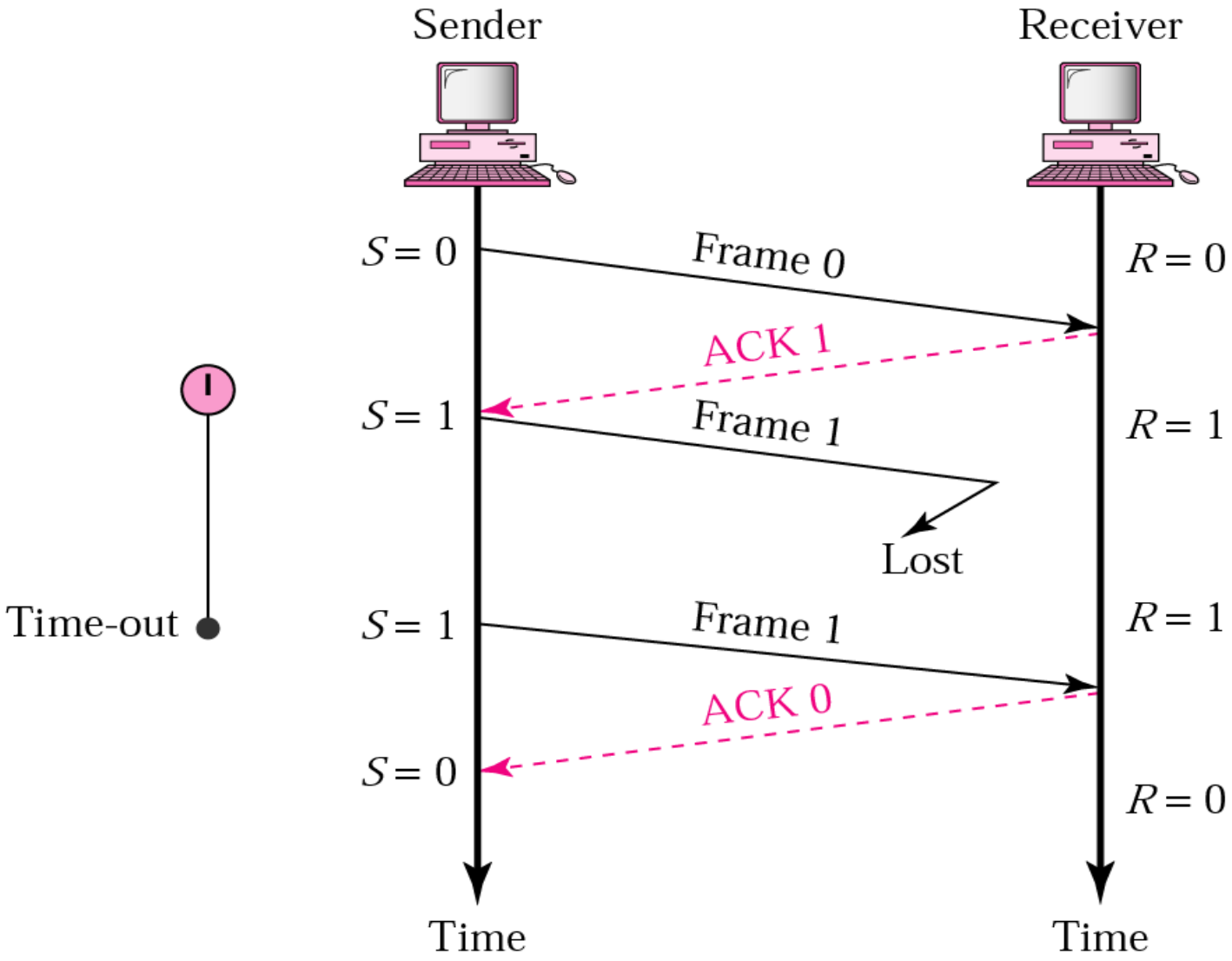
ວິທີນີ້ຝ່າຍສົ່ງຈະສົ່ງເຟຣມຂໍ້ມູນມາໃຫ້ໜຶ່ງເຟຣມ ແລະ ລໍການຕອບ Acknowledge (ACK) ຈາກຝ່າຍຮັບເມື່ອຝ່າຍຮັບໄດ້ຮັບການຕອບຮັບ ACK ຈາກຝ່າຍຮັບກະປຽບເໝືອນການຕອບຮັບ “OK” ວ່າໄດ້ຮັບແລ້ວຝ່າຍສົ່ງກໍ່ຈະທຳການສົ່ງຕໍ່ໄປ.



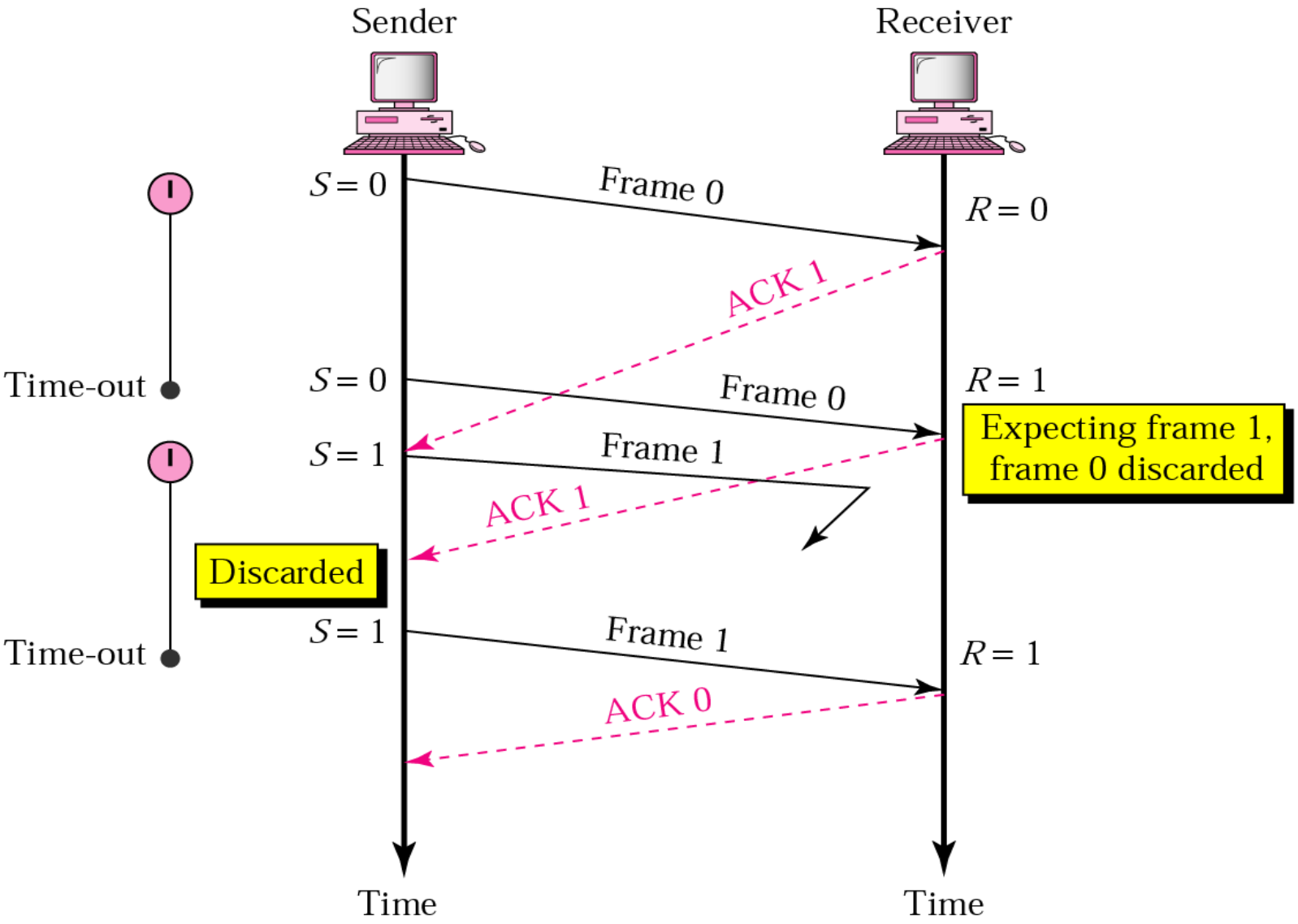
Stop-and-Wait , lost ACK frame



Stop-and-Wait , lost frame

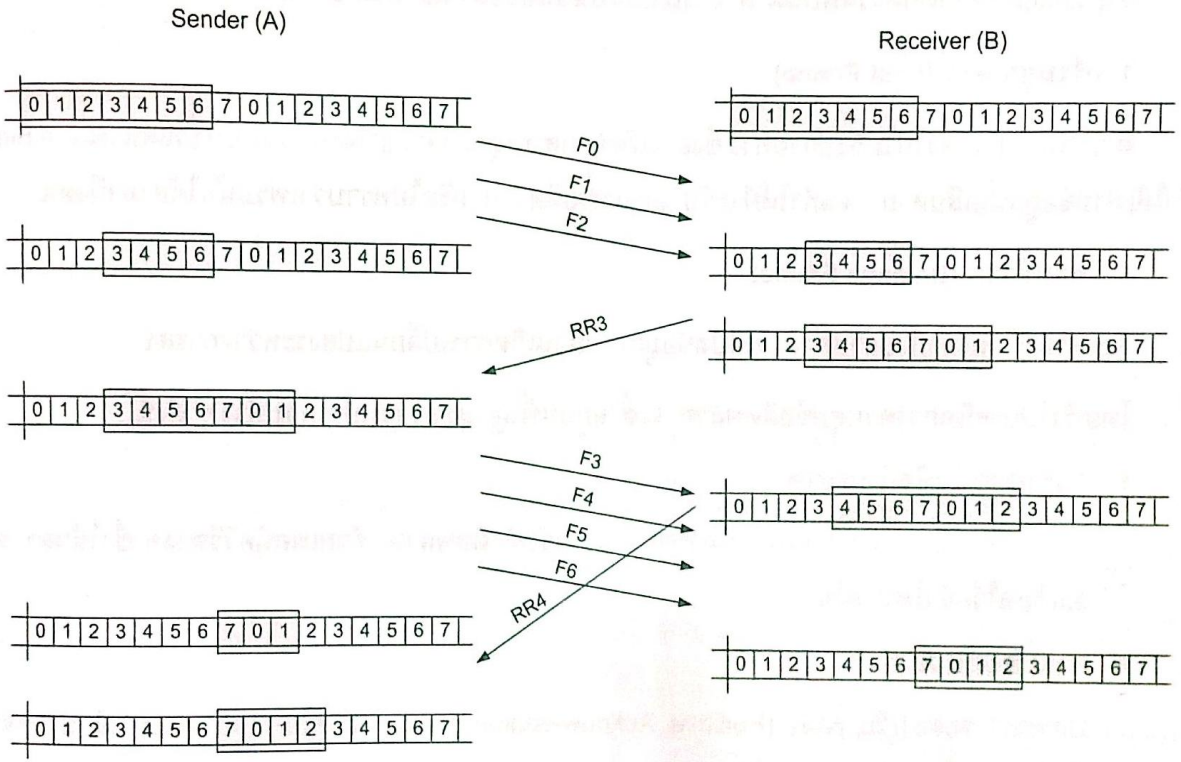


Stop-and-Wait , delayed ACK and lost frame

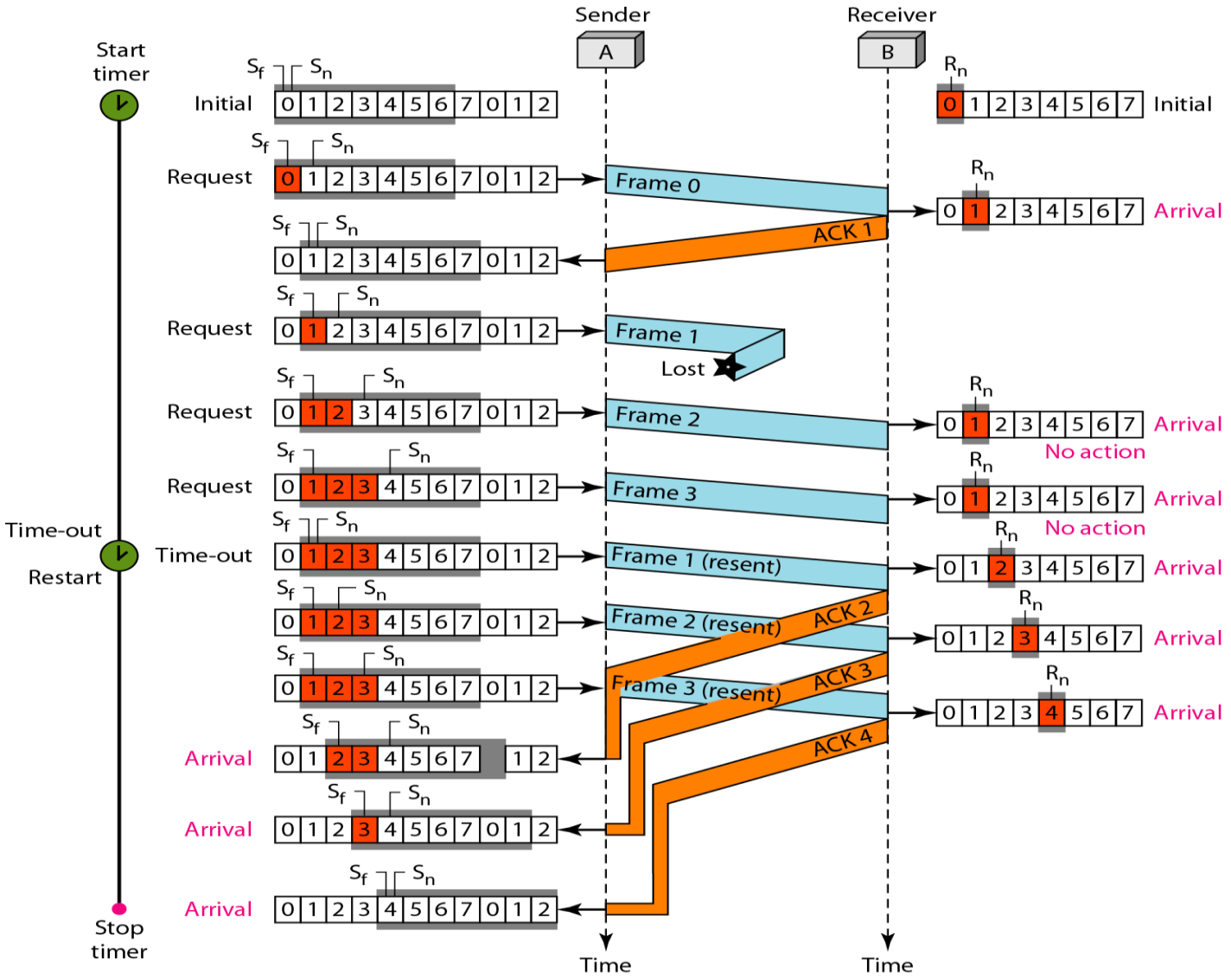


3.1.2 ການຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນດ້ວຍວິທີເລື່ອນໜ້າຕ່າງ (Sliding-Window Flow Control)

ວິທີຄວບຄຸມການໄຫຼຂອງຂໍ້ມູນແບບເລື່ອນໜ້າຕ່າງນີ້ຝ່າຍສົ່ງສາມາດສົ່ງເຟຣມຂໍ້ມູນຫຼາຍໆເຟຣມກ່ອນທີ່ຈະໄດ້ຮັບການຕອບຮັບຄືນ ຝ່າຍຮັບຈະມີການຕອບຮັບກັບໄປພຽງບາງເຟຣມເທົ່ານັ້ນ ດັ່ງນັ້ນ ການຕອບຮັບ ACK ໃນໜຶ່ງຄັ້ງຈະໝາຍເຖິງການໄດ້ຮັບເຟຣມມາແລ້ວຫຼາຍເຟຣມນັ້ນເອງ ເຊິ່ງເປັນວິທີທີ່ດີກວ່າວິທີທຳອິດ.



Lost frame


Sender
Receiver
A
B
Start timer
Initial
Request
Frame 0
ACK 1
Request
Frame 1
Lost
Request
Frame 2
No action
Request
Frame 3
No action
Time-out
Restart
Time-out
Frame 1 (resent)
Frame 2 (resent)
ACK 2
Frame 3 (resent)
ACK 3
ACK 4
Arrival
Arrival
Arrival
Arrival
Arrival
Arrival
Arrival
Stop timer
Time
Time

3.2 ການຄວບຄຸມຂໍ້ຜິດພາດ (Error Control)

ສຳລັບການຄວບຄຸມຂໍ້ຜິດພາດນີ້ຈະດຳເນີນກ່ຽວກັບ

- ຈະຕ້ອງກວດສອບຂໍ້ຜິດພາດຂອງເຟຣມແນວໃດ ແລະຈະຕ້ອງເຮັດແນວໃດຫາກເກີດຂໍ້ຜິດພາບຂຶ້ນ
- ເມື່ອຝ່າຍຮັບໄດ້ທຳການກວດຈັບຂໍ້ຜິດພາດຂອງຂໍ້ມູນທີ່ສົ່ງມາຝ່າຍຮັບສາມາດດຳເນີນການຄວບຄຸມຂໍ້ຜິດພາດທີ່ເກີດຂຶ້ນໄດ້ໃນ 3 ກໍລະນີດ້ວຍກັນຄື:

1. ບໍ່ຕ້ອງດຳເນີນການໃດໆ (Do Nothing)
2. ຕອບກັບຂໍ້ຄວາມໄປຍັງຝ່າຍສົ່ງ (Return a Message) ເພື່ອໃຫ້ຝ່າຍສົ່ງທຳການສົ່ງຂໍ້ມູນສ່ວນທີ່ເສຍຫາຍມາຮອບໃໝ່ (Retransmitted)
3. ກວດແກ້ຂໍ້ຜິດພາດ (Correct the Error)

ສຳລັບຂໍ້ຜິດພາດທີ່ກວດພົບນັ້ນສາມາດແບ່ງອອກໄດ້ 2 ຊະນິດດ້ວຍກັນຄື:

1. ເຟຣມສູນຫາຍ (Lost Frame)
2. ເຟຣມຖືກທຳລາຍ (Damage Frame)

ໂດຍທົ່ວໄປເທັກນິກການຄວບຄຸມຂໍ້ຜິດພາດຈະຕັ້ງຢູ່ສ່ວນປະກອບຕ່າງໆ ດັ່ງນີ້:

- *ການກວດຈັບຂໍ້ຜິດພາດ*
- *ການຕອບຮັບ ACK*
- *ການສົ່ງຂໍ້ມູນໃນຮອບໃໝ່ຫຼັງຈາກລໍຖ້າຈົນຄົບເວລາ (Timeout)*
- *ການຕອບຮັບ NAK ແລະສົ່ງຂໍ້ມູນໃນຮອບໃໝ່*