Tietoliikennespesifikaation mallintaminen ja analyysi

Tässä kotitehtävässä mallinnetaan DECT (Digital European Cordless Telephone) systeemin virheitä havaitsevaa siirtoprotokollaa (unilateral jump). Unilateral jump perustuu alternating bit protokollaan, mutta on sitä hieman laajempi.

Alternating bit protokolla on yksinkertainen. Jokainen datapaketti on numeroitu (modulo 2). Protokolla perustuu siihen, että uutta pakettia ei ryhdytä lähettämään ennen kuin vanha on kuitattu. Esimerkiksi "lähettäjä" ei voi lähettää pakettia 1 ennen kuin vastaanottaja on kuitannut paketin 0. (Huomaa, että tässä "lähettäjä" tarkoittaa DATAN lähettäjää. Tiedonsiirron täytyy toimia kuitenkin myös toiseen suuntaan. Tämä siksi, koska "vastaanottajan" täytyy kuitata datapaketteja.)

Unilateral jump lisää tähän protokollaan yhden uuden piirteen. Jos "vastaanottaja" ei lähetä positiivista kuittausta esim. paketista 0 (mahd. monen uudelleenlähetyksenkään jälkeen), voi "lähettäjä" spontaanisti siirtyä lähettämään pakettia 1. Tällöin paketin 0 yli "hypätään" mikä selittänee protokollan nimen.

Seuraavassa osassa "lähettäjän" ja "vastaanottajan" toiminta on kuvattu ETSIn dokumenttiin (prETS 300175-3) perustuen.

Receiver:

The three state variables at the receiver are:
• LAST-BCK meaning "what packet number was transmitted in the last BCK bit to indicate the number of the next expected I_P segment";
• LAST-PKT meaning "what I_P packet number appeared in the last databurst with correct received A-field and containing I_P data";
• THIS-PKT meaning "what I_P packet number appears in the databurst just received with correct received A-field and containing I_P data".

The receiver, after evaluating the A-CRC of the received I_P packet evaluates the state variables following table 1 to know how the transmitter acted; after that, it evaluates the Rb-CRCs of all the B subfields and shall act as indicated in table 2.

THIS-PKT=LAST-PKT	THIS-PKT=LAST-BCK	How transmitter acted	Resulting case
yes	yes	retransmit	(a)
no	yes	normal advance	(b)
yes	no	unnecessary retransmit	(c)
no	no	jump	(d)

Table 1

Table 1 State	Rb-CRC result	how Receiver shall act
(a)	Rb-CRC passed	LAST-BCK=(LAST-BCK+1) MOD 2; RX requires a new I_P packet
(a)	Rb-CRC failed	LAST-BCK=LAST-BCK; RX requires the retransmission of the I_P packet
(b)	Rb-CRC passed	LAST-BCK=(LAST-BCK+1) MOD 2; RX requires a new I_P packet
(b)	Rb-CRC failed	LAST-BCK=LAST-BCK; RX still requires the same I_P packet
(c)	Rb-CRC passed	LAST-BCK=LAST-BCK; RX still requires the same I_P packet
(c)	Rb-CRC failed	LAST-BCK=LAST-BCK; RX still requires the same I_P packet
(d)	Rb-CRC passed	LAST-BCK=LAST-BCK; RX requires a new I_P packet (alignment to TX jump); The I_P packet just received is accepted
(d)	Rb-CRC failed	LAST-BCK=(LAST-BCK+1) MOD 2; RX requires the retransmission of the I_P packet (alignment to TX jump); The I_P packet just received is rejected

Table 2

Transmitter:

Define: LTI_P=packet number of the last transmitted I_P segment.
There are two state variables at the transmitter:
- $\Sigma ACKN$ meaning "has the receiver acknowledged the LTI_P at least once?"; For duplex bearer (as in this case) the LTI_P is acknowledged by receiving a packet with $Q2$ bit set to 1. *Note:* $\Sigma ACKN$ is reset to "no" if the transmission of a new I_P packet starts.
- $LAST-BCK$ meaning "what was the setting of the last correct received BCK bit (with the BCK bit the receiver reports the next expected I_P packet)?"; *Note:* If $\Sigma ACKN$ switches to "yes" also a new BCK is received.

On receipt of new segments, the transmitter shall look at the values of its state variables and act according to table 3 In state (b) the transmitter can choose between retransmission and jump. Its choice is reflected in the pkt number chosen. If jump the pkt number toggles, if retransmit the pkt number is unchanged.

Case	$\Sigma ACKN$	LTI_P=LAST-BCK	How transmitter shall act
(a)	yes	no	normal advance or retransmit
(b)	yes	yes	retransmit/jump
(c)	no	no	retransmit
(d)	no	yes	retransmit/jump

Table 3

Tehtävä

URL:issa `http://www.tcs.hut.fi/Teaching/Tik-79.179/ujump.net` on edelläolevaan spesifikaatioon perustuva (osittainen) PROD-malli. Täydentäkää malli (siitä puuttuu 12 transitiota). Tehkää mallille saavutettavuusanalyysi, ja varmistakaa, ettei siinä esiinny lukkiumia. Palauttakaa täydellinen PROD-malli, ja laittakaa vastaukseenne myös saavutettavuusgraafin statistiikka (solmujen ja kaarien lukumäärä). Statistiikan saatte proben komennolla `statistics`.

Vinkkejä

- Kaikki vastaanottajapuolen paikat ja transitiot sisältävät etuliitteen rx_.
- Vastaanottajapuolen tilamuuttujien vastaavuudet ovat: LAST-BCK=rx_last_bck, LAST-PKT=rx_last_pkt. Tilamuuttujaa THIS-PKT ei olla mallinnettu, koska datapaketin numero on suoraan luettavissa viestistä.
- Transitiot rx_case_?_detect laittavat 5-tuppelin 'ether' paikasta johonkin vastaanottajan lokaaliin paikkaan Table 1:n testiin perustuen; transitiot rx_case_?_send lähettävät joko positiivisen tai negatiivisen kuittauksen tulleeseen viestiin (Table 2:n tapaukset).
- Lähettäjän paikat ja transitiot sisältävät nimessään etuliitteen tx_.
- Lähettäjäpuolen tilamuuttujien vastaavuudet ovat: $\Sigma ACKN$=tx_acks, LTI_P=tx_last_tx_packet, ja LAST-BCK=tx_last_bck.
- Lähettäjäpuolen transitiot tx_detect_acks_prev_acks, ja tx_detect_acks_no_prev_acks päivittävät paikalliset tilamuuttujat. Ensimmäinen transitio hoitaa tapauksen, missä tuleva paketti on jo ennestään kuitattu (positiivisesti). Toinen transitio puolestaan käsittelee tapahtuman, missä tulevaa pakettia ei ole kuitattu.
- Virheitä kuvataan lähettäjän mahdollisuudella lähettää joko virheetön tai virheellinen CRC.
- Tässä mallissa on oletettu kännykän (PP=Portable Part) olevan 'lähettäjä' ja tukiaseman (FP=Fixed Part) olevan 'vastaanottaja'.
- 'ether' paikka mallintaa ilmarajapintaa lähettäjän ja vastaanottajan välillä. Se sisältää 5-kenttäisen tuppelin. Ensimmäinen kenttä sisältää paketin numeron. Toinen kertoo, kuka lähetti ko. viestin: mobiili (TRANSMITTER=PP) vai tukiasema (RECEIVER=FP). Kolmannesta kentästä löytyy bck-bitti, mikä ilmaisee seuraavan paketin, mitä vastaanottaja odottaa. Neljäs kenttä pitää sisällään Rb-CRC:n eli tarkastussumman. Virheetöntä tarkastussummaa esitetään numerolla 1, ja virheellistä numerolla 0. Viimeinen kenttä sisältää Q2-bitin. Jos datapaketti tulee virheittä perille, vastaanottaja kuittaa sen lähettämällä takaisin viestin, missä Q2-bitti on 1. Muussa tapauksessa vastataan viestillä, missä Q2-bitti on 0 (ks. transitio rx_case_a_send).

Huom! Kotitehtävä on itsenäinen harjoitus. Yhteistyö ei saa näkyä vastauspaperissasi.
Palautuspäivä on perjantai 26.5.2000.
Palautus tapahtuu sähköpostilla. Elektroninen palautus on tehdään ASCII:na, PostScriptinä ja HTML:nä (lisätietoja ja lisämuotovaatimuksia assistentilta).
Kotitehtäväassistentti on Annikka Aalto (sähköposti: `Annikka.Aalto@hut.fi`.)