Chapter 7: Z-transform analysis of protocols

1. Z-transform - Applied to probability distributions


Definition:

— \( F(z) = \sum_{n=0}^{\infty} f_n z^n \)
— some applications use \( z^n \) others use \( z^{-n} \).
— applications in probability in particular use \( z^n \) because the derivatives yield the moments of the distributions.
The transform is also known as the moment generating function of a discrete distribution.
— If \( f_n = P(x = n) \) for a discrete distribution, then
  \[
  F(z) = \sum_{n=0}^{\infty} P(x = n)z^n
  \]
  \[
  F(1) = \sum_{n=0}^{\infty} P(x = n) = 1 \text{ since the sum of the probabilities must } = 1.
  \]
— Important properties:
  — The z-xform for each probability distribution is unique
    If we have z-transform, we can find the probability distribution
    Inverting from the transform domain to the probability function is often difficult
    There are tables of inverse functions
  — We can find the moments without finding the probability distribution
    \[
    \frac{dF(z)}{dz} \bigg|_{z=1} = \sum_{n=0}^{\infty} nP(x = n) = E(x)
    \]
    Higher order moments are found by taking higher order derivatives
    \[
    \frac{d^2 F(z)}{dz^2} \bigg|_{z=1} = \sum_{n=0}^{\infty} n(n - 1)P(x = n) = E(x^2) - E(x)
    \]
  — Final value theorem: \( \lim_{z \to 1} (1 - z)F(z) = f_{\infty} \)
2. Z-transform applied to Markov Processes

Markov Processes are similar to FSM’s, in that they have states with transitions between them. The transitions are probabilistic, rather than deterministic.

The transitions are labeled with the probability of taking a transition. We will also label the transitions with a cost, such as the delay or message transmission.

For instance, the composite FSM of the retransmission protocols can be written as a Markov Process, where the probability of a message transmission has a probability of occurrence of 1, but the probability of a message being received is \(1 - P_E\), and the probability of the message being lost is \(P_E\).

We can also assign the cost of a message transmission as \(N_M\) when a message is transmitted and \(N_A\) when an acknowledgement is transmitted. Or we can assign the cost of a transition as the delay that the transition incurs. For instance, if the transition corresponds to the transmission of a message the delay is the transmission, propagation and queuing delays, or if the transition occurs when there is a timeout, the delay is the retransmission period.

We will use z-transforms to determine the moments of the cost, or the complete distribution function of the delay, to move between specific states. For instance the mean and variance of the delay to transmit a message in the retransmission protocols. One way to perform the analysis is to use the final value theorem.

2.1 The z-transform of the Markov Process

1. \(P_i(j)\) = the probability of being in state \(i\) at the \(j^{th}\) step

The z-transform of the \(i^{th}\) state is defined as:

\[
P_i(z) = \sum_{j=0}^{\infty} P_i(j)z^j
\]

2. The relationship between the \((j-1)^{th}\) and \(j^{th}\) state is:

\[
P_i(j) = \begin{cases} U(i) & j = 0 \\ \sum_k P_{ik}P_k(j-1) & j > 0 \end{cases}
\]

Where

\[
U(i) = \begin{cases} 1 & \text{if } i \text{ is the initial state} \\ 0 & \text{otherwise} \end{cases}
\]

\(P_{ik}\) = the probability of moving to state \(i\) from state \(k\)

the label of the arc from state \(k\) to state \(i\)

3. The z-transforms of the states are inter-related by

\[
P_i(z) = \sum_{j=0}^{\infty} P_i(j)z^j
\]

\[
= U(i) + \sum_{j=1}^{\infty} \sum_k P_{ik}P_k(j-1)z^j
\]

\[
= U(i) + \sum_k P_{ik} \sum_{j=1}^{\infty} P_k(j-1)z^j
\]

\[
= U(i) + z \sum_k P_{ik} \sum_{j=0}^{\infty} P_k(j)z^j
\]

\[
= U(i) + z \sum_k P_{ik}P_k(z)
\]

Note that shifting one step back corresponds to multiplying the z-transform by \(z\).
A shift of one step is sufficient to describe the relationship of a Markov process, because the current state is only dependent on the previous state.

It is possible to define more complex processes, where the current state is dependent on the previous "n" state, as in difference equations. In general, \( \sum_{j=k}^{\infty} P_j(j-k)z^j = z^k \sum_{j=0}^{\infty} P_j(j)z^j. \)

4. In an Markov Process with M states, there are M equations for the M states:

We can find the z-transform of each state by solving a set of simultaneous equations.

Define

\[ \vec{P}(z) = [P_1(z), P_2(z), \ldots, P_M(z)] \]
\[ \vec{U} = [U(1), U(2), \ldots, U(M)], \]

where \( \vec{U} = [1, 0, \ldots, 0], \) if 1 is the initial state

\[ T = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1M} \\ P_{21} & P_{22} & \cdots & P_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M1} & P_{M2} & \cdots & P_{MM} \end{bmatrix} \]

\[ \vec{P}^T(z) = \vec{U}^T + zT \vec{P}^T(z), \] and

\[ \vec{P}^T(z) = (I-zT)^{-1} \vec{U}^T \]
2.2 Finding the cost to get between 2 states:

1. Draw a Markov Process with the first state as the initial state.
2. Replace the exiting transitions from the final state with a self-loop, so that the machine does not progress beyond that state.
3. Assign a cost to each transition.
   The cost can be the number of bits transmitted, the delay, the amount of processing, ...
4. Define a cumulative cost function as:

\[
C(j) = \begin{cases} 
0 & j = 0 \\
C(j-1) + \sum_k P_k(j-1) \sum_i C_{ik} P_{ik} & j > 0 
\end{cases}
\]

with a z-transform

\[
C(z) = \sum_{j=0}^{\infty} C(j) z^j
\]

5. Manipulating the z-transform,

\[
C(z) = \sum_{j=1}^{\infty} C(j) z^j = \sum_{j=1}^{\infty} C(j) z^j \\
= \sum_{j=1}^{\infty} (C(j-1) z^j + \sum_k P_k(j-1) z^j \sum_i C_{ik} P_{ik}) \\
= zC(z) + z \sum_k P_k(z) \sum_i C_{ik} P_{ik} \\
= \frac{z}{1-z} \sum_k P_k(z) \sum_i C_{ik} P_{ik}
\]

6. To find the total cost, we apply the final value theorem:

\[
\lim_{z \to 1} (1-z) C(z) = \sum_k P_k(1) \sum_i C_{ik} P_{ik} = \tilde{P}(1) \tilde{C}_P^T
\]

where \( \tilde{C}_P = [\sum_i C_{i1} P_{i1}, \sum_i C_{i2} P_{i2}, \cdots, \sum_i C_{iM} P_{iM}] \)

is the vector of the average costs that we incur in each state.
This technique calculates the cost without inverting the z-transform.

General technique:

- Draw all of the arcs between the 2 states
- Use perturbation analysis on component machines to get a composite machine
- Start the analysis on the first state - this is the initial state
- Force a deadlock on the final state by inserting a self loop that occurs with probability 1
- The first and final state can be the same state, but should be distinguished in the composite machine
- Assign a probability and cost to each arcs
- Use the z-transform to calculate the average cost to get between each the two states
### 2.3 Example: a simple message transfer protocol.

- Uniquely identified messages and acknowledgements.
- The round trip delay channel delay is less than the timeout retransmission interval, so that retransmissions only occur when messages are lost because of an error.
- The terminated machine corresponds to a single message transfer.

Where:

(Src, FC, Rcv, RC) = the states of the component machines

- SRC = the source,
  - 1=Xmit
  - 0= Wait for Ack

- FC = the forward channel from the source to the receiver
  - M= Message on channel
  - E = Empty

- RCV = the receiver, and
  - 0 = waits for message
  - 1 = xmit Ack

- RC = the reverse channel from the receiver to the source.
  - A = Ack on the channel
  - E = Empty

There is at most one message on the forward and reverse channels because the round trip delay is greater than the interval between retransmission attempts.

The channels deliver a message with probability $P_C$, and lose the message with probability $1 - P_C$.

The cost to transmit a message is $C_M$, and the cost to transmit an acknowledgement is $C_A$.

By inspection of the composite machine:
\[
T = \begin{bmatrix}
0 & 1 - P_C & 0 & 1 - P_C & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & P_C & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & P_C & 1
\end{bmatrix}
\]

and,
\[
\bar{C}_p = [C_M, 0, C_A, 0, 0]
\]

Therefore,
\[
\bar{P}^T(z) = (1-zT)^{-1}\bar{U}^T = \frac{1}{1-z^2(1-P_C) - z^4P_C(1-P_C)} \begin{bmatrix}
1 \\
z \\
z^2P_C \\
z^3P_C \\
z^4P_C
\end{bmatrix}
\]

and
\[
\bar{P}(1) = \begin{bmatrix}
1 \\
\frac{1}{P_C} \\
\frac{1}{P_C^2} \\
\frac{1}{P_C} \\
1
\end{bmatrix}.
\]

The average cost to deliver the message is:
\[
\lim_{z \to 1} (1-z)C(z) = \bar{P}(1)\bar{C}_p = \frac{C_M}{P_C} + \frac{C_A}{P_C}
\]

Delays
Calculate the delay until the message is received the first time.
The final state is state 3.
The cost function is the delay on the links.
The cost of transition 1,2 is 0,
the cost of transition 2,3 is \(t_f\), a random variable that is the delay on the forward channel, and
the cost of transition 2,1 is \(T\), the time between retransmissions.
the average delay is \( T/P_C + E(t_f) \).

**Homework**

1. **Z-transform of a protocol**

   Consider the following model of a protocol

   ![Diagram of a protocol model]

   The links are labeled with the probability of making the transition

   \( P_i(j) = \) the probability of being in state \( i \) on the \( j^{th} \) of the protocol operation

   \[
   P_i(z) = \sum_{j=0}^{\infty} P_i(j)z^j
   \]

   State 1 is the initial state of the protocol

   A. Write the transition matrix \( T \) of the protocol

   B. Write the set of simultaneous equations that relate the \( P_i(z) \)

2. For the state machine shown, use \( z \)-transforms to determine the average number of transitions for a machine that starts in state 1 at \( j = 0 \) to arrive at state 3.

   ![Diagram of a state machine]

   a. \( P_s = \) ?

   b. Write \( P_1(j) \), \( P_2(j) \) and \( P_3(j) \) for \( j \geq 0 \).

   c. Write \( P_1(z) \) as a function of \( P_2(z) \) and \( z \), \( P_2(z) \) as a function of \( P_1(z) \) and \( z \), and \( P_3(z) \) as a function of \( P_2(z) \) and \( z \).
d. Solve $P_3(z)$ as a function of $z$ only.

e. Find the average value of $j$ that the machine reaches state 3.

3. Extra Credit.

Determine the $z$-transform for the stop and wait protocol on a full duplex channel with both the messages and acknowledgements numbered modulo 2.

Can you determine the transform from the 4 component processes, without constructing the composite machine.

I haven’t done this example, but it may be possible to do this by recognizing that the interaction of the machines is a convolution, which is a multiplication of the matrices in the transform domain.
3. M/M/1 Queues: A derivation using z-transforms

$$P_N(z) = \sum_{i=0}^{\infty} p_i z^i$$

where $p_i$ is the probability of being in state $i$

$$1 - \lambda \delta = 1 - \lambda \delta - \mu C \delta$$

From the diagram of the Markov chain

$$p_0 = (1 - \lambda \delta) p_0 + \mu C \delta p_1$$

$$\lambda p_0 = \mu C p_1$$

$$\rho p_0 = p_1$$

$$(1 + \rho) p_0 = p_0 + p_1$$

$$p_k = (1 - \lambda \delta - \mu C \delta) p_k - \lambda \delta p_{k+1} + \mu C \delta p_{k+1}$$ for $k \geq 1$

$$(1 + \rho) p_k = \rho p_{k+1} + p_k$$

Summing the terms

$$(\rho + 1) \sum_{k=0}^{\infty} p_k z^k = \rho \sum_{k=0}^{\infty} p_{k-1} z^k + \mu C \delta \sum_{k=0}^{\infty} p_{k+1} z^k + p_0 + p_1$$

$$= \rho \sum_{k=0}^{\infty} p_k z^k + \frac{1}{z} \sum_{k=0}^{\infty} p_k z^k - \frac{1}{z} (p_0 + z p_1) + p_0 + p_1$$

$$= \rho \sum_{k=0}^{\infty} p_k z^k + \frac{1}{z} \sum_{k=0}^{\infty} p_k z^k + (1 - \frac{1}{z}) p_0$$

$$(1 + \rho) P_N(z) = \rho z P_N(z) + \frac{1}{z} P_N(z) + (1 - \frac{1}{z}) p_0$$

$$P_N(z) = \frac{1}{1 - \rho z} p_0$$

$$P_N(1) = 1 = \frac{p_0}{1 - \rho}$$

$$p_0 = 1 - \rho$$

$$N = \left. \frac{dP_N(1)}{dz} \right|_{z=1} = \frac{\rho}{1 - \rho}$$

From Little’s Law

$$T = \frac{N}{\lambda} = \frac{\rho}{\lambda (1 - \rho)} = \frac{1}{\mu C (1 - \rho)}$$

The waiting time is

$$W = \frac{1}{\mu C (1 - \rho)} = \frac{\rho}{\mu C (1 - \rho)}$$

The number in queue is

$$N_Q = \lambda W = \frac{\rho^2}{1 - \rho}$$

REFERENCES