Définitions

Sûreté de fonctionnement

- Aptitude à délivrer un service de confiance justifiée

Service délivré par un système : son comportement tel que perçu par son, ou ses utilisateurs

Utilisateur : autre système en interaction avec le système considéré

Fonction d'un système : ce à quoi le système est destiné, décrite par la spécification fonctionnelle

Défaillance de la sûreté de fonctionnement

Défaillance (du service) : (failure) événement lorsque le service délivré dévie du service correct, soit il n'est plus conforme à la spécification, soit la spécification ne décrit pas de manière adéquate la fonction du système

Erreur : (error) l'état du système susceptible d'entraîner une défaillance

Faute : (fault) cause adjugée ou supposée d'une erreur

Fautes (**Occurences**) Erreurs (**Propagation**) Défaillances (**Conséquences**) Fautes ...

Sûreté de fonctionnement

- Aptitude à éviter des défaillances du service plus fréquentes ou plus graves qu' acceptable

Fiabilité

- Reliability
- Définie par ITU (International Telecommunications Unions) recommendation E.800 :

- Définition (Reliability,Fiabilité)

The ability of an item to perform a required function under given conditions for a given time interval.

Aptitude d'un système à accomplir une fonction spécifiée (exigée) durant un interval de temps dans des conditions données

  - donc prendre en compte un intervalle de temps
  - le système doit être opérationnel pendant toute la période

Disponibilité

- Availability
- Egalement définie par une recommendation ITU

- Définition (Availability,Disponibilité)

  The ability of an item to be in a state to perform a required function at a given instant of time assuming that the external resources, if needed, are provided.

  Propriété à délivrer correctement le service demandé (en terme de délai et de qualité) à un instant donné.

  - donc prendre en compte une date
  - le système doit être opérationnel à cette date.

Sûreté de fonctionnement

- **Attributs :**
  - Disponibilité : prêt à l'utilisation
  - Fiabilité : continuité du service
  - Sûreté (safety) : absence de conséquences catastrophiques pour l'environnement
  - Confidentialité (confidentiality) : absence de divulgations non-autorisées de l'information
  - Intégrité (integrity) : absence d'altérations inappropriées du système
  - Maintenabilité (maintenability) : aptitude aux réparations et aux évolutions
- **Moyens :**
  - Prévention de fautes
  - Tolérance aux fautes
  - Elimination des fautes

Sécurité (security) : Absence d'accès ou de manipulations non-autorisés de l'état du système
Disponibilité + Confidentialité + Intégrité

Application

- Application traditionnelles : aviation, espace, industrie
  safety/life critique 6nines disponibilité : % 99.9999
- Applications récentes : E-commerce, finance, télécommunications, réservation avion
  non safety/life critique 5nines disponibilité : % 99.999
- Applications scientifiques non critique
- Applications à disponibilité usuelle (4nines) : mobile

Si le système fonctionne, étude de ses performances

- Evaluation quantitative des mesures considérées (débit, temps de réponse, etc.)

Evaluation Quantitative

- Mesures
  - plus de précision mais coût élevé
  - impossible pendant la phase de développement
  - utilisation des techniques statistiques

- Evaluation à partir d'un modèle
  - abstraction du système réel
  - modèle mathématique en fonction des paramètres
  - méthodes de résolutions
    - simulation à événements discrets
    - résolution analytique

Etapes de modélisation

- Construction du modèle
  (par des méthodes de spécification de haut niveau)
- Détermination de paramètres du modèle
  (par mesures, méthodes statistiques)
- Résolution du modèle
- Interprétation des résultats
- Validation du modèle

Phénomènes aléatoires :

- Taux d'arrivée des requêtes, attaques, tâches
- Temps d'exécution, transmission, traitement
- Taux de panne, réparation

Modèles probabilistes
Compromis entre la précision et la complexité d'analyse

Détermination de paramètres
Mesures et méthodes statistiques

Modèles

- Processus stochastiques (Chaînes de Markov)
- Modèles statiques (Arbres de Fautes, Reliability Bloc Diagram (RBD)
- File d'attentes, Graphe de précedences, ...

Méthodes

- Simulation
- Résolution numérique
- Solution analytique, Solution à forme produit

Diagramme de Fiabilité (Reliability Bloc Diagram (RBD))

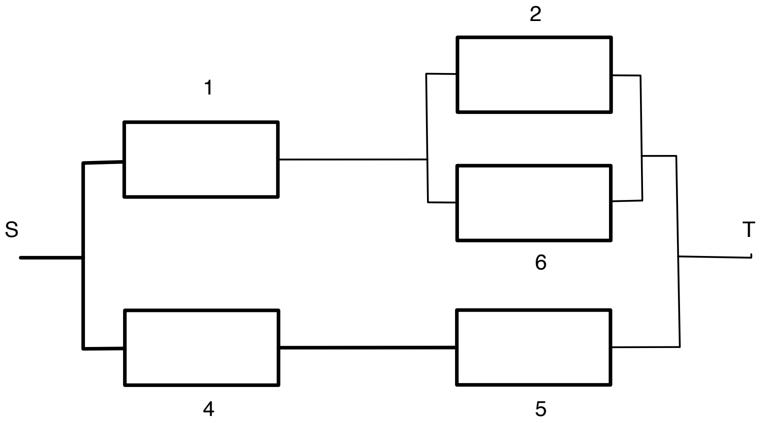
- Une représentation graphique du système et de la fiabilité.
- Chaque composant est représenté par un bloc.
- Sert à déterminer si le système est UP ou DOWN en fonction des états des composants.
- Idée intuitive : un bloc peut être vu comme un switch qui est fermé quand le composant est UP et ouvert quand le composant est DOWN.
- Il y a une entrée dans le diagramme S et une sortie T.

RBD

- C'est un modèle reposant sur la logique et non pas sur les états.
- Modèle Statique : pas de représentation du temps ni de l'ordre entre des événements successifs.
- Hypothèse d'Indépendence des pannes des différents composants.
- Pas de pannes arrivant conjointement ou de pannes provoquées par la panne d'un autre composant.

RBD

- Le système est UP si il y a au moins un chemin passant par des éléments UP et reliant S à T.



Logique

- Le comportement du système par rapport à la panne est modélisé par les connexions entre blocs.
- Si tous les composants sont nécéssaires, les modéliser en série
- Si un seul des composants est nécéssaire, les modéliser en parallèle.
- Si il en faut au moins K parmi N, utiliser la structure "K out of N"

Blocs en Série

- n composants indépendants en série.
- E_i le composant i fonctionne.
- $R_s = P(E_1 \cap E_2 \cap \ldots \cap E_n)$
- A cause de l'indépendance :

$$P(E_1 \cap E_2 \cap \ldots \cap E_n) = \prod_{i=1}^{n} P(E_i)$$

- En notant $R_i = P(E_i)$, on obtient :

$$R_s = \prod_{i=1}^{n} R_i$$

- On remarque que $R_s < min(R_i)$. Le système est moins fiable que sa composante la moins fiable.

Blocs en Parallèle

- n composants indépendants en prallèle.
- E_i le composant i fonctionne.
- $R_p = P(E_1 \cup E_2 \cup \ldots \cup E_n)$
- Le système est en panne si tous les composants sont en panne :

$$1 - R_p = \prod_{i=1}^{n} (1 - R_i)$$

Systèmes Série-Parallèles

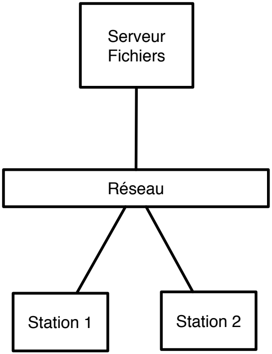
- Décomposition récursive : un système série-parallèle (SP) est soit :
  - un bloc isolé
  - plusieurs sous-systèmes SP en série
  - plusieurs sous-systèmes SP en parallèle
- Utilise la décomposition récursive de la construction pour obtenir la fiabilité.
- Exemple simple : n étages en série, chaque étage composé de m composants en parallèle tous identiques :

$$R_{sp} = (1 - (1 - R)^m)^n$$

Exemple : Station de Travail/Serveur de Fichiers

- Un serveur de fichiers,
- Deux stations de Travail identiques
- Un réseau pour les connecter. On suppose que le réseau est fiable.
- Le système est opérationnel si le serveur de fichiers est opérationnel et au moins une des deux stations de travail est opérationnelle.

Représentation de l'exemple



Etude de la fiabilité de l'exemple

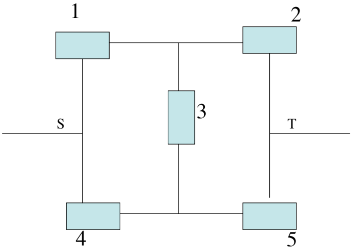
- R_w fiabilité d'une station de travail
- R_f fiabilité du serveur de fichiers
- R_{fsw} fiabilité du système

$$R_{fsw} = (1 - (1 - R_w)^2)R_f$$

Systèmes NON Série-Parallèles

- Un graphe est SP ssi il ne contient ni structure N ni structure W.
- Si on ne peut plus utiliser la décomposition SP, on peut enumérer et construire la table Booléenne.

réseau avec bridge :



Utilisation de la table Booléenne

- Examiner tous les cas UP et DOWN pour tous les composants
- Dans chaque cas, évaluer si le système est UP ou DOWN
- Calculer les probabilités de chaque cas (facile, c'est le produit des probas élémentaires à cause de l'hypothèse d'indépendance).
- Exemple : Si $E1 = E2 = E4 = 1$ et $E3 = E5 = 0$ le système est UP et la probabilité de cette configuration est $R_1 R_2 R_4 (1 - R_3)(1 - R_5)$.
- Sommer les probabilités que le système soit UP

Première partie de la table de l'exemple

1	2	3	4	5	Bridge	Probability
0	0	0	0	0	0	
0	0	0	0	1	0	
0	0	0	1	0	0	
0	0	0	1	1	1	(1-R1)(1-R2)(1-R3)R4R5
0	0	1	0	0	0	
0	0	1	0	1	0	
0	0	1	1	0	0	
0	0	1	1	1	1	(1-R1)(1-R2)R3R4R5
0	1	0	0	0	0	
0	1	0	0	1	0	
0	1	0	1	0	0	
0	1	0	1	1	1	(1-R1)R2(1-R3)R4R5
0	1	1	0	0	0	
0	1	1	0	1	0	
0	1	1	1	0	1	(1-R1)R2R3R4(1-R5)
0	1	1	1	1	1	(1-R1)R2R3R4R5

Seconde partie de la table de l'exemple

1	0	0	0	0	0	
1	0	0	0	1	0	
1	0	0	1	0	0	
1	0	0	1	1	1	R1(1-R2)(1-R3)R4R5
1	0	1	0	0	0	
1	0	1	0	1	1	R1(1-R2)R3(1-R4)R5
1	0	1	1	0	0	
1	0	1	1	1	1	R1(1-R2)R3R4R5
1	1	0	0	0	1	R1R2(1-R3)(1-R4)(1-R5)
1	1	0	0	1	1	R1R2(1-R3)(1-R4)R5
1	1	0	1	0	1	R1R2(1-R3)R4(1-R5)
1	1	0	1	1	1	R1R2(1-R3)R4R5
1	1	1	0	0	1	R1R2R3(1-R4)(1-R5)
1	1	1	0	1	1	R1R2R3(1-R4)R5
1	1	1	1	0	1	R1R2R3R4(1-R5)
1	1	1	1	1	1	R1R2R3R4R5

Résultat pour l'exemple

- En agrégeant la table précédente et en factorisant on trouve que :

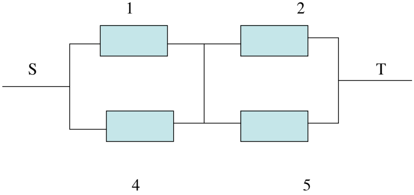
$$
\begin{aligned}
R_{bridge} &= R_1 R_2 \\
&+ R_1(1-R_2)(R_4R_5 + R_3(1-R_4)R_5) \\
&+ (1-R_1)R_4(R_5 + (1-R_5)R_2R_3)
\end{aligned}
$$

Conditionnement et Factorisation

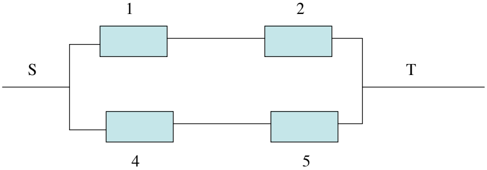
- Mais il faut considérer les 2^n configurations si il y a n objets.
- On conditionne sur le l'état d'un composant (ou de plusieurs composants) pour se ramener à des structures déjà étudiées ou faciles (SP)
- Sur l'exemple du bridge, on conditionne sur l'état du bloc 3.

Conditionnement sur 3

3 fonctionne



3 ne fonctionne pas



Conditionnement

- Si le composant 3 est DOWN, on obtient un modèle SP
- Si le composant 3 est UP, on obtient également un modèle SP
- On applique le théorème de conditionnement et les formules pour les modèles série-parallèles.

- $R_{3down} = 1 - (1 - R_1 R_2)(1 - R_4 R_5)$
- $R_{3up} = (1 - (1 - R_1)(1 - R_2))(1 - (1 - R_4)(1 - R_5))$
- On applique le théorème de conditionnement

$$R_{bridge} = R_3 R_{3up} + (1 - R_3) R_{3down}$$

Composant K parmi N

- Système consistant en N composants indépendants.
- Le système est UP quand K ou plus de ces composants sont UP.
- Cas Identique : tous les composants ont le même taux de panne et de réparation.
- Cas Non Identique : Les composants ont des taux de panne et de réparation distincts par composant.

Avec sous-composants identiques

- Soit R la fiabilité d'un composant.
- On additionne les probabilités de toutes les configurations avec au moins K composants opérationnels.

$$R(K,N) = \sum_{j=K}^{N} R^j (1-R)^{N-j} \frac{N!}{j!(N-j)!}$$

- Somme partielle d'une distribution binomiale.
- Algorithme classique (attention aux approximations numériques)

Récurrence dans le cas général

- Système dont les composants sont distincts. La fiabilité du composant i est R_i.
- 3 remarques simples :
  - Un système avec $K = 0$ (sans contraintes) est toujours UP.

$$R(0, N) = 1$$

  - Un système trop contraint est toujours DOWN.

$$R(j, N) = 0 \quad si\ j > N$$

  - En conditionnant sur l'état du composant N :

$$R(i, N) = (1 - R_N)R(i, N - 1) + R_N R(i - 1, N - 1)$$

- Algorithme récursif.