

INDEX

- Abstractions, total order broadcast, 33–34
- Accreditation, safety critical systems, 267
- Adaptive meta-heuristics (routing), 525–562
 - genetic algorithms, 532–536
 - capacity assignment, 534–535
 - fault tolerance, 534
 - multicast routing, 533–534
 - unicast routing, 533
 - genetic routing protocol, implementation, 547–552
 - generally, 547–548
 - genetic parameters, 551–552
 - links, 548–549
 - nodes, 549–551
 - topologies, 548
 - traffic, 551
 - genetic routing protocol design, 536–547
 - algorithm overview, 539–540
 - chromosome encodings and evaluation, 536–537
 - comparisons, 541–544
 - execution flow, 544–545
 - fault tolerance, 545–547
 - path genetic operators, 537–539
 - overview, 525–526
 - results, 552–560
 - delay comparisons, 557
 - fault tolerance, 559–560
 - overhead comparisons, 557–559
 - variable crossover, 555–557
 - variable delay, 553
 - variable mutation, 553–555
 - routing problem, 526–532
 - current methods, 530–532
 - design characteristics, 528–529
 - least-cost algorithms, 527
- AES cryptography, 576–579
- Agents (software), critical infrastructures, 483–485. *See also* Critical infrastructures
- Agreement, total order broadcast, 32, 34
- Algorithms. *See also* Adaptive meta-heuristics (routing)
 - design fault tolerance, 231–232
 - diagnosis, system-level diagnosis, 157–160, 160–164
 - total order broadcast, 34–36
 - weak interactive consistency, 40–42
- Analytical availability modeling, 287
- Application-specific techniques, design fault tolerance, 231–232
- Approval voting, voting schemes, 97–98
- Approximate availability model, VAXcluster processing subsystem, availability modeling case study, 303–304
- Availability-based composite availability, availability modeling, 292–297
- Availability modeling, 285–317
 - approaches, 286–292
 - analytical, 287
 - generally, 286–287
 - non-state space models, 287–288
 - state space models, 289–292
 - case study (Digital Equipment Corporation), 297–315
 - fault tree, 300–301
 - generally, 297–298
 - reliability block diagram (RBD), 299–300
 - SPN model, 309–315
 - VAXcluster processing subsystem, 301–304
 - VAXcluster storage subsystem, 304–308

- Availability modeling (*Continued*)
 - composite availability, 292–297
 - availability-based, 292–294
 - performance- and availability-based, 294–297
 - performance-based, 294
 - overview, 285–286
- Backup paths, network resilience, 454–455
- Bandwidth reallocation, telemedical disaster relief, H3M architecture, 363
- Benchmarking, dependability evaluation, 340–342
- “Best-effort” primary/backup paths, network resilience, 468–473
 - case study, 472–473
 - costs and detestation, 470–472
 - policy driven design, 468–470
- Binary decision diagram (BDD), formal verification techniques, 3–4, 5–7
- Biological tissue simulation, WIRM system architecture, 355
- Block diagrams, model-based evaluation, methodologies and tools, 62
- Byzantine failure model, state machine replication, 31–32
- Candidate of internal equivalent pairs (CIEP), combinational circuit equivalence, 10–12
- Capacity assignment, adaptive meta-heuristics (routing), 534–535
- Centralized diagnosis algorithms, system-level diagnosis, 160–162
- Certification, weak interactive consistency, 38–39
- Channel service time in dimension, Laplace-Stieltjes transform computation of, traffic self-similarity, 510–512
- Chromosome encodings, adaptive meta-heuristics (routing), genetic routing protocol design, 536–537
- CMOS technology, IBM mainframe dependable computing, error detection, 378, 379–380
- Combinational circuit equivalence, 7–14
 - approaches to, 7–9
 - formal verification techniques, false negatives, 12–13
 - internal equivalence points, 9–12
 - summarization, 13–14
- Commerical-off-the-shelf (COTS) hardware environment, integrated reliable real-time systems, integration issues, 427–428
- Communication energy, wireless sensor networks, 118
- Compilation, safety critical systems, 254
- Component structured fault tolerance, 232–236
- Composite availability (availability modeling), 292–297
 - availability-based, 292–294
 - performance- and availability-based, 294–297
 - performance-based, 294
- Computational integrity, IBM mainframe dependable computing, 372–373
- Computational tree logic, model checking and, sequential circuits, 22–23
- Computer-aided design, field programmable gate arrays (FPGAs), 603
- Computer vision, WIRM system architecture, 355
- Continuous time Markov chains:
 - model-based evaluation, 62–63
 - VAXcluster processing subsystem, availability modeling case study, 301–303
- Control channels, wireless sensor networks, 118
- Cooperating-monitor-based modalities, predicate detection in asynchronous systems, faulty environments, 203–209
- Cooperating monitors, predicate detection in asynchronous systems, faulty environments, 190–191
- Correctness concerns, voting, 104–106
- Costs:
 - of failures, 244
 - network resilience, “best-effort” primary/backup paths, 470–472
 - safety critical systems, 252–253
 - wireless sensor networks, 120
- Critical infrastructures, 479–499
 - categories of, 480–481
 - electrical supply, 482–483

- future prospects, 497
- overview, 479–480
- Safeguard architecture, 486–497
 - electrical supply, 492–497
 - generally, 486–488
 - generic agents, 488
 - telecommunications, 489–492
- solutions, 483–486
 - agents (software), 483–485
 - detection, 485
 - response, 486
 - telecommunications, 481–482
- Cross-entropy pathfinding, network
 - resilience, 460–468
 - generally, 460–461
 - implementation, 466–468
 - initialization and selection strategies, 465–466
 - method, 461–462
 - mobile agents, 462–465
- Cryptography, 563–595. *See also*
 - Field programmable gate arrays (FPGAs)
 - AES cryptography, 576–579
 - case study (twofish cipher), 579–589
 - described, 579–582
 - key-schedule mapping, 585–588
 - mapping, 583–585
 - performance analysis, 588–589
 - overview, 563–565
 - reconfigurable computing, 565–576
 - applications, 573–576
 - FPGAs, 567–568
 - future of, 589–590
 - generally, 565–566
 - history of, 566–567
 - systems, 568–573
- Data channels, wireless sensor
 - networks, 118
- Data diversity, voting, data fusion, 101–102
- Data error propagation (software), 395–417
 - analysis, 408–409
 - attributes and characteristics, 414–415
 - injection, 407–408
 - overview, 395–396
 - results, 409–414
 - setup, 401–407
 - description files, 407
 - environment simulators and test cases, 404–405
 - error injection triggering, 403
 - error types and injection locations, 402–403
 - faults and fault triggers, 401–402
 - generally, 401
 - logging variables, memory areas, and events, 403–404
 - target system instrumentation, 405–406
 - target system model, 396–397
 - tool suite, 397–401
 - system structure, 397–399
 - work process, 399–401
- Data fusion (voting), 98–102
 - components, 100
 - data diversity, 101–102
 - examples, 101
 - sensor processing, 99
- Datagram routing protocols, adaptive
 - meta-heuristics (routing), 531–532
- Data integrity, IBM mainframe dependable
 - computing, 372
- Dependability. *See also* Model-based
 - evaluation
 - design fault tolerance, 222–225
 - safety critical systems, 247–250
 - benchmarking, 340–342
 - fault injection, 323–337
 - accuracy, 332–337
 - components, attributes, and properties, 324–327
 - generally, 323–324
 - technologies, 327–331
 - hardware implemented, 327–328
 - hybrid tools and Xception approach, 329–331
 - simulation implemented, 328
 - software implemented, 328–329
 - field measurement, 321–323
 - analysis, 322–323
 - data collection, 321–322
 - overview, 319–320
 - robustness testing, 337–340
 - application, 338–339
 - correction, 339–340
 - generally, 337–338
- Design fault tolerance, 213–241
 - benefits, 225–230
 - adoption of methods, 228–230
 - models and evidence, 226–228
 - potential, 225–226
 - secondary, 228
 - design solutions, 230–236

- Design fault tolerance (*Continued*)
 - algorithm- and application-specific techniques, 231–232
 - component structured, 232–236
 - generic techniques, 231
 - examples and principles, 215–225
 - components, 215–219
 - dependability goals, 222–225
 - diversity, 220–222
 - redundancy and failure diversity, 219–220
 - overview, 213–215
- Deterministic stochastic petri nets,
 - model-based evaluation, 66
- Diagnosability algorithms, system-level diagnosis, 157–160
- Diagnosis algorithms, system-level diagnosis, 160–164
 - adaptively diagnosable systems, 162
- Digital Equipment Corporation, availability modeling case study, 297–315. *See also* Availability modeling
- Dimension-ordered routing, k -ary n -cube and, traffic self-similarity, 504–506
- Distance vector routing, adaptive meta-heuristics (routing), 530
- Distributed algorithms, system-level diagnosis, diagnosis algorithms, 162–164
- Distributed computations, predicate detection in asynchronous systems, fault-tree environments, 173–174
- Diversity, design fault tolerance, 219–222
- DMNA protocol operation, telemedical disaster relief, H3M architecture, 361–362
- Documentation, safety critical systems, 254–255
- Echo broadcast, weak interactive consistency, 38
- Education, safety critical systems, 267
- Electrical supply, critical infrastructures, 482–483, 492–497
- Emergence, network resilience, 457–460. *See also* Network resilience
- Energy balance, wireless sensor networks, 135
- Equipment representation, safety critical systems, 259
- Equivalence checking:
 - of finite state machine (FSM), sequential circuits, 19–21
 - formal verification techniques, 5
- Error confinement, component structured fault tolerance, 233–234
- Error correction, component structured fault tolerance, 234–235
- Error detection:
 - component structured fault tolerance, 233–234
 - IBM mainframe dependable computing, 375–380
 - CMOS technology, 378
 - failure modes, 375–376
 - fault locating tests, 376–377
 - inline error checking, 378–379
 - objectives (system/360), 376
 - TCM technology, 377–378
- Executions, component structured fault tolerance, 235
- Extended lattice state, predicate detection in asynchronous systems, faulty environments, 186–188
- Extended observation system, predicate detection in asynchronous systems, faulty environments, 184
- Failure detection, predicate detection in asynchronous systems, 177–183
- Failure modes, IBM mainframe dependable computing, 375–376
- Failure tolerance. *See* State machine replication
- False negatives, combinational circuit equivalence, 12–13
- Fault-affected computations, predicate detection in asynchronous systems, faulty environments, 188
- Fault containment, integrated reliable real-time systems, 425–426
- Fault-free network, wireless sensor networks, 121–123, 137–139
- Fault injection, dependability evaluation, 323–337
 - components, attributes, and properties, 324–327
 - generally, 323–324
 - technologies, 327–331
 - hardware implemented, 327–328
 - hybrid tools and Xception approach, 329–331

- simulation implemented, 328
- software implemented, 328–329
- Fault locating tests, IBM mainframe
 - dependable computing, 376–377
- Fault model, wireless sensor networks, 120
- Fault removal:
 - figures of merit and assumptions, 77
 - maintenance strategies, 76–77
 - model, 77–80
 - numerical evaluation, 80–82
- Fault taxonomy, field programmable gate arrays (FPGAs), 603–608
- Fault tolerance. *See also* Design fault tolerance; Memory fault tolerance; State machine replication
 - adaptive meta-heuristics (routing), 534, 545–547
 - results, 559–560
- Fault-tolerant permutation routing, wireless sensor networks, 123–124
- Fault-tree environments, predicate
 - detection in asynchronous systems, 173–177
 - distributed computations, 173–174
 - observation system, 175
 - observer dependence, 175–177
 - predicate properties, 175
 - problem in, 174–175
- Fault trees:
 - availability modeling, non-state space models, 288
 - availability modeling case study, 300–301
 - model-based evaluation, methodologies and tools, 61–62
- Faulty environments, predicate detection in asynchronous systems, 183–194
 - classifications, 193–194
 - detection modalities, 186–191
 - detection semantics, 191–193
 - extended observation system, 184
 - impact of failure detection, 184–185
 - operational state of processes, 183–184
 - predicate properties, 185–186
 - solutions, 194–209
 - cooperating-monitor-based modalities, 203–209
 - impossibility result, 195–196
 - local-state-lattice-based modalities, 202–203
 - perfect predicate detection, 196–199
 - stabilizing variants, 199–202
- Faulty network, wireless sensor networks, 132–135
 - energy balance, 135
 - permanent faults, 132
 - transient faults, 132–135
- Field programmable gate arrays (FPGAs), 597–625
 - failures, 608–621
 - detection mechanisms, 610–617
 - configuration-based testing, 612
 - functional testing, 612–615
 - on-line/off-line testing, 612
 - parametric testing, 615–617
 - test patterns generation, 611–612
 - generally, 608–609
 - recovery approaches, 617–621
 - terminology, 609–610
 - fault taxonomy, 603–608
 - future trends, 621–622
 - overview, 597–598
 - preliminaries, 598–602
 - computer-aided design, 603
 - logic block architecture, 600–601
 - programming technology, 598–600
 - routing architecture, 601–602
 - reconfigurable computing, 567–568
- Figures of merit and assumptions, model-based evaluation, 69–70, 77
- Filtering, weak interactive consistency, 40
- Finite state machine (FSM):
 - equivalence checking of, sequential circuits, 19–21
 - sequential circuits, 14–15
- Flooding, network resilience, 455
- Formal safety critical systems methods, 244–245, 250–252. *See also* Safety critical systems
- Formal verification techniques, 3–25
 - binary decision diagram (BDD), 5–7
 - combinational circuit equivalence, 7–14
 - approaches to, 7–9
 - false negatives, 12–13
 - internal equivalence points, 9–12
 - summarization, 13–14
 - generally, 4–5
 - overview, 3–4
 - sequential circuits, 14–24
 - computational tree logic and model checking, 22–23
 - finite state machine (FSM), 14–15
 - equivalence checking of, 19–21
 - generally, 14
 - reachable state representation, 15–19
 - summarization, 23–24

- Generic fault tolerance techniques, 231
- Genetics algorithms. *See* Adaptive meta-heuristics (routing)
- Granularity, component structured fault tolerance, 235–236
- Hardware, voting, dependable systems, 89–90
- Hardware faults, dependability evaluation, fault injection accuracy, 332–333
- Hardware-implemented fault injection, dependability evaluation, 327–328
- Hot pluggable channels, IBM mainframe dependable computing, online repair, 388
- H3M architecture, telemedical disaster relief, 359–366
 - bandwidth reallocation and sharing, 363
 - DMNA protocol operation, 361–362
 - generally, 359–361
 - intra- and intercluster communication, 362–363
 - specialization in, 364–366
- Human-computer interface, safety critical systems, 255
- IBM mainframe dependable computing, 369–393
 - error detection and fault isolation, 375–380
 - CMOS technology, 378, 379–380
 - failure modes, 375–376
 - fault locating tests, 376–377
 - inline error checking, 378–379
 - objectives (system/360), 376
 - TCM technology, 377–378
 - founding concepts, 370–372
 - historical perspective, 369–370
 - implementation, 375
 - instruction level retry, 380–386
 - concept stage, 380
 - ILR challenges, 382–385
 - ILR in CMOS, 385
 - implementation (initial), 381–382
 - permanent CPU failures, 385–386
 - online repair, 386–391
 - channel subsystem, 388
 - CPU concurrent repair, 389
 - memory fault tolerance, 389–391
 - power and cooling, 388
 - service processor, 387
 - special purpose system/360 MPs, 386
 - system/370 commercial MPs, 386–387
 - principles, 372–374
- Implicit spares, field programmable gate arrays (FPGAs) failure, 618–620
- Impossibility results:
 - predicate detection in asynchronous systems, faulty environments, 195–196
 - voting, 103–104
- Incremental validation, integrated reliable real-time systems, 428–429
- Inline error checking, IBM mainframe dependable computing, error detection, 378–379
- Instruction level retry, IBM mainframe dependable computing, 380–386
 - concept stage, 380
 - ILR challenges, 382–385
 - ILR in CMOS, 385
 - implementation (initial), 381–382
 - permanent CPU failures, 385–386
- Integrated reliable real-time systems, 419–447
 - application (aerospace), 432–442
 - implementation validation, 441–442
 - integration issues, 435–437
 - RMS-application synchronization, 440–441
 - software architecture, 437–439
 - strong partitioning, 439–440
 - system architecture, 432–435
 - background, 421–425
 - fault taxonomy, 423–425
 - terminology, 421–423
 - integration issues, 425–429
 - COTS hardware environment, 427–428
 - hybrid redundancy levels and types, 426
 - incremental validation, 428–429
 - interapplication communication, 427
 - legacy software support, 428
 - resource sharing, 426–427
 - strong partitioning, 425–426
 - overview, 419–421
 - progress in, 429–432
 - architectural support, 430–431
 - scheduling, 431
 - strong partitioning, 429–430
 - synchronization, 431–432
- Integrity, total order broadcast, 32

- Interactive remote visualization, telemedical disaster relief, 358–359
- Interconnection network. *See* Traffic self-similarity
- Internal equivalence points, combinational circuit equivalence, 9–12
- K -ary n -cube, dimension-ordered routing and, traffic self-similarity, 504–506
- Laplace-Stieltjes transform computation, Channel service time in dimension, traffic self-similarity, 510–512
- Least-cost algorithms, adaptive meta-heuristics (routing), 527. *See also* Adaptive meta-heuristics (routing)
- Legacy software, integrated reliable real-time systems, 428
- Links, adaptive meta-heuristics (routing), genetic routing protocol, 548–549
- Link state routing, adaptive meta-heuristics (routing), 530–531
- Local predicates (predicate detection in asynchronous systems):
 - fault-tree environments, 175
 - faulty environments, 185
- Local state lattice, predicate detection in asynchronous systems, faulty environments, 188–190, 202–203
- Logic block architecture, field programmable gate arrays (FPGAs), 600–601
- Low-power control channel, wireless sensor networks, 125–131
 - described, 126–128
 - performance, 128–131
 - variant in packet transfer latency, 131
- LSI design, combinational circuit equivalence, 7
- Markov chain:
 - availability modeling, state space models, 289–290
 - continuous time, VAXcluster processing subsystem, availability modeling case study, 301–303
- Markovian models, model-based evaluation, 62–65
- Markov regenerative stochastic petri nets, model-based evaluation, 66
- Markov rewards, availability modeling, state space models, 290–291
- Mean message latency, calculation of, traffic self-similarity, 515
- Mean waiting time at source node, calculation of, traffic self-similarity, 515
- Medical image analysis modules, WIRM system architecture, 355
- Memory fault tolerance, IBM mainframe dependable computing, online repair, 389–391
- Message blocking in dimension, computation of, traffic self-similarity, 512–513
- Meta-heuristics. *See* Adaptive meta-heuristics (routing)
- Miter, combinational circuit equivalence, 9
- Model-based evaluation, 57–86
 - design decisions, 68–76
 - α -count evaluation, 74–76
 - α -count mechanism, 68–69
 - α -count model, 71–74
 - figures of merit and assumptions, 69–70
 - fault removal, 76–82
 - figures of merit and assumptions, 77
 - maintenance strategies, 76–77
 - model, 77–80
 - numerical evaluation, 80–82
 - methodologies, 61–66
 - combinatorial techniques, 61–62
 - Markovian models, 62–65
 - non-Markovian models, 65–66
 - overview, 57–58
 - role of, 58–61
 - tools, 67–68
- Model checking:
 - computational tree logic and, sequential circuits, 22–23
 - formal verification techniques, 5
- Multicast routing, adaptive meta-heuristics (routing), 533–534
- Multichannel network, wireless sensor networks, 135–139
 - fault-free network, 137–139
 - system model, 135–136
- Muteness failure detector, 44–52
 - implementation, 46–49
 - interactions, 49–52
 - model, 44–45
 - specification, 45–46
 - weak interactive consistency, 37

- Network channels, source nodes and, traffic self-similarity, 508–510
- Network resilience, 449–477
 - “best-effort” primary/backup paths, 468–473
 - case study, 472–473
 - costs and detestation, 470–472
 - policy driven design, 468–470
 - comparisons, 455–456
 - cross-entropy pathfinding, 460–468
 - generally, 460–461
 - implementation, 466–468
 - initialization and selection strategies, 465–466
 - method, 461–462
 - mobile agents, 462–465
 - design parameters, 450–452
 - discussed, 473–475
 - emergence, 457–460
 - overview, 449–450
 - protection, 452–453
 - reconfiguration, 453–454
 - self-healing, 454–455
 - span versus end-to-end reestablishment, 452
- Network size, wireless sensor networks, 117–118
- Nodes, adaptive meta-heuristics (routing), genetic routing protocol, 549–551
- Non-state space models, availability modeling, 287–288
- Observation system, predicate detection in asynchronous systems, fault-tree environments, 175
- Observer dependence, predicate detection in asynchronous systems, fault-tree environments, 175–177
- Observer independence, predicate detection in asynchronous systems, faulty environments, 185
- Off-line testing, field programmable gate arrays (FPGAs), 612
- Online repair (IBM mainframe dependable computing), 374, 386–391
 - channel subsystem, 388
 - CPU concurrent repair, 389
 - memory fault tolerance, 389–391
 - power and cooling, 388
 - service processor, 387
 - special purpose system/360 MPs, 386
 - system/370 commercial MPs, 386–387
- On-line testing, field programmable gate arrays (FPGAs), 612
- Operational state of processes, predicate detection in asynchronous systems, faulty environments, 183–184
- Packet transfer latency, variant in, low-power control channel, wireless sensor networks, 131
- Performance-based composite availability, availability modeling, 294–297
- Performance concerns, voting, 106–107
- Permanent faults, wireless sensor networks, 120, 132
- Permutation routing, in single-hop networks, wireless sensor networks, 121–125
- Plurality voting, voting schemes, 96–97
- Predicate detection in asynchronous systems, 171–212
 - failure detection, 177–183
 - fault-tree environments, 173–177
 - distributed computations, 173–174
 - observation system, 175
 - observer dependence, 175–177
 - predicate properties, 175
 - problem in, 174–175
 - faulty environments, 183–194
 - classifications, 193–194
 - detection modalities, 186–191
 - detection semantics, 191–193
 - extended observation system, 184
 - impact of failure detection, 184–185
 - operational state of processes, 183–184
 - predicate properties, 185–186
 - solutions, 194–209
 - cooperating-monitor-based modalities, 203–209
 - impossibility result, 195–196
 - local-state-lattice-based modalities, 202–203
 - perfect predicate detection, 196–199
 - stabilizing variants, 199–202
 - overview, 171–173
- PROPANE. *See* Data error propagation (software)
- Reachable state representation, sequential circuits, 15–19

- Real-time systems. *See* Integrated reliable real-time systems
- Reconfigurable computing, 565–576.
See also Cryptography; Field programmable gate arrays (FPGAs)
 applications, 573–576
 FPGAs, 567–568
 future of, 589–590
 history of, 566–567
 systems, 568–573
- Recovery blocks, component structured fault tolerance, 234–235
- Reduced ordered binary decision diagram (ROBDD), combinational circuit equivalence, 8
- Redundancy, design fault tolerance, 219–220
- Reliability, model-based evaluation, 58–61
- Reliability block diagram (RBD):
 availability modeling, non-state space models, 287–288
 availability modeling case study, 299–300
- Reliable multicast, total order broadcast, 33
- Reliable real-time systems. *See* Integrated reliable real-time systems
- Rerouting, network resilience, 455
- Risk (safety critical systems), 247–253
 costs, 252–253
 dependability, 247–250
 formal methods, 250–252
- RMS-application synchronization, integrated reliable real-time systems, 440–441
- Robust communication primitives.
See Wireless sensor networks
- Robustness testing (dependability evaluation), 337–340
 application, 338–339
 correction, 339–340
 generally, 337–338
- Routing, dimension-ordered, k -ary n -cube and, traffic self-similarity, 504–506. *See also* Adaptive meta-heuristics (routing)
- Routing algorithms. *See* Traffic self-similarity
- Safeguard system. *See* Critical infrastructures
- Safety. *See* Critical infrastructures
- Safety critical systems, 243–271
 applications, 253–256
 compilation, 254
 complementary methods, 255
 design, 254
 documentation, 254–255
 human-computer interface, 255
 requirements, 253–254
 standards, 256
 static analysis, 255–256
 testing, 256
 cost of failure, 244
 formal methods, 244–245
 education and accreditation, 267
 generally, 265–266
 research into, 266
 standards, 267–268
 technology, 266–267
 historical perspective, 247
 overview, 243–244
 risk, 247–253
 costs, 252–253
 dependability, 247–250
 formal methods, 250–252
 specification framework, 256–262
 equipment representation, 259
 safety requirements classification, 259–260
 safety requirements specification, 260–262
 state changes, 256–259
 specifications, 245–246
 system state and behavior, 262–265
 railway points, 264–265
 railway signals, 263–264
 train tracks, 262–263
- Security. *See* Critical infrastructures
- Self-similarity. *See* Traffic self-similarity
- Semi-Markov stochastic petri nets, model-based evaluation, 65–66
- Sensor processing, voting, data fusion, 99
- Sequential circuits, 14–24
 computational tree logic and model checking, 22–23
 finite state machine (FSM), 14–15
 equivalence checking of, 19–21
 generally, 14
 reachable state representation, 15–19
 summarization, 23–24
- Service centric model (wireless sensor networks), 279–283
 compared, 282–283
 described, 280–282
 generally, 279–280

- Set-decreasing predicates, predicate
 - detection in asynchronous systems, faulty environments, 186
- Simulation-based fault injection,
 - dependability evaluation, 328
- Software:
 - model-based evaluation, 67–68
 - voting, dependable systems, 90–92
- Software data error propagation, 395–417.
 - See also* Data error propagation (software)
- Software design fault tolerance.
 - See* Design fault tolerance
- Software faults, dependability evaluation,
 - fault injection accuracy, 333–336
- Software-implemented fault injection,
 - dependability evaluation, 328–329
- Source nodes, network channels and, traffic
 - self-similarity, 508–510
- SPN model, availability modeling case study, 309–315
- Stabilizing variants, predicate detection in asynchronous systems, faulty environments, 199–202
- Stable predicates, predicate detection in asynchronous systems:
 - fault-tree environments, 175
 - faulty environments, 185
- Standards, safety critical systems, 256
- State changes, safety critical systems, 256–259
- State correction, component structured
 - fault tolerance, 234–235
- State machine replication, 27–56.
 - See also* Fault tolerance
 - background, 28–29
 - motivations, 27–28, 52
 - muteness failure detector, 44–52
 - implementation, 46–49
 - interactions, 49–52
 - model, 44–45
 - specification, 45–46
 - rationale, 29–30
 - related approaches, 30–31
 - system model, 31–32
 - Byzantine failure model, 31–32
 - execution and communication, 31
 - total order broadcast, 32–36
 - abstractions, 33–34
 - algorithm composition, 34–36
 - specification, 32–33
 - weak interactive consistency, 36–44
 - abstractions, 37–40
 - algorithm, 40–42
 - certificates, 42–44
 - generally, 36–37
- State space models (availability modeling), 289–292
 - Markov chains, 289–290
 - Markov rewards, 290–291
 - stochastic petri nets and stochastic reward nets, 291–292
- Static analysis, safety critical systems, 255–256
- Stochastic petri nets:
 - model-based evaluation, 63–66
 - stochastic reward nets, availability modeling, state space models, 291–292
- Strong partitioning, integrated reliable real-time systems, 425–426, 429–430, 439–440
- Surface Point Signature, 3D data
 - compression technique, telemedical disaster relief, 356–358
- Systemic failures. *See also* Design fault tolerance
 - defined, 213
 - sources of, 213–214
- System integrity, IBM mainframe
 - dependable computing, 372–373
- System-level diagnosis, 143–169
 - applications, 165–166
 - classification of systems, 148–157
 - adaptively diagnosable systems, 157
 - excess-diagnosable systems, 153–155
 - incrementally diagnosable systems, 156–157
 - partially diagnosable systems, 152–153
 - sequentially diagnosable systems, 155–156
 - uniquely diagnosable systems, 150–152
 - described, 145–148
 - diagnosability algorithms, 157–160
 - diagnosis algorithms, 160–164
 - adaptively diagnosable systems, 162
 - centralized, 160–162
 - distributed algorithms, 162–164
 - new approaches, 164
 - large and complex systems, 143–145

- Telecommunications, critical
 - infrastructures, 481–482, 489–492
- Telemedical disaster relief, 349–368
 - current status, 350–352
 - H3M architecture, 359–366
 - bandwidth reallocation and sharing, 363
 - DMNA protocol operation, 361–362
 - generally, 359–361
 - intra- and intercluster communication, 362–363
 - specialization in, 364–366
 - interactive remote visualization, 358–359
 - overview, 349–350
 - 3D data compression technique, 356–358
 - WIRM system architecture, 352–355
 - challenges, 354–355
 - generally, 352–354
- Termination, total order broadcast, 34
- Testing, safety critical systems, 256
- Thermal Conduction Modules (TCM)
 - technology, IBM mainframe dependable computing, error detection and fault isolation, 377–379
- 3D data compression technique,
 - telemedical disaster relief, 356–358
- Threshold voting, voting schemes, 95–96
- Time histories, state changes, safety critical systems, 256–259
- Time synchronization, wireless sensor networks, 119
- Topologies, adaptive meta-heuristics (routing), genetic routing protocol, 548
- Total order, total order broadcast, 32
- Total order broadcast, 32–36. *See also*
 - State machine replication
 - abstractions, 33–34
 - algorithm composition, 34–36
 - specification, 32–33
 - state machine replication, 27–28
- Traffic, adaptive meta-heuristics (routing), genetic routing protocol, 551
- Traffic self-similarity, 501–524
 - analytical model, 507–518
 - channel service time in dimension, Laplace-Stieltjes transform computation of, 510–512
 - generally, 507
 - mean message latency, calculation of, 515
 - mean waiting time at source node, calculation of, 515
 - message blocking in dimension, computation of, 512–513
 - source nodes and network channels, 508–510
 - validation of, 515–518
 - virtual channel occupancy
 - probabilities in dimension, computation of, 513–514
 - k -ary n -cube and dimension-ordered routing, 504–506
 - modeling of, 506–507
 - notations, 523–524
 - overview, 501–504
 - routing performance, 518–519
- Transient faults, wireless sensor networks, 120, 132–135
- Transition costs, wireless sensor networks, 118–119
- Twofish cipher:
 - cryptography, 579–589
 - described, 579–582
 - key-schedule mapping, 585–588
 - mapping, 583–585
 - performance analysis, 588–589
- Unicast routing, adaptive meta-heuristics (routing), 533
- Uninterrupted applications, IBM mainframe dependable computing, 373–374
- Validation, incremental, integrated reliable real-time systems, 428–429
- Validity, total order broadcast, 32, 34
- VAXcluster:
 - processing subsystem, availability
 - modeling case study, 301–304
 - SPN model, availability modeling case study, 309–315
 - storage subsystem, availability modeling case study, 304–308
- Virtual channel occupancy probabilities
 - in dimension, computation of, traffic self-similarity, 513–514
- Voting, 87–114
 - data fusion, 98–102
 - components, 100
 - data diversity, 101–102
 - examples, 101
 - sensor processing, 99

- Voting (*Continued*)
 - dependable systems, 88–94
 - hardware, 89–90
 - software, 90–92
 - weighted voting framework, 92–94
 - field programmable gate arrays (FPGAs)
 - failure, 620–621
 - implementation, 102–107
 - correctness concerns, 104–106
 - impossibility results, 103–104
 - performance concerns, 106–107
 - overview, 87–88
 - schemes, 94–98
 - approval voting, 97–98
 - plurality voting, 96–97
 - taxonomy, 94–95
 - threshold voting, 95–96
 - unifying concepts, 107–110
 - data-centered methodology, 109–110
 - terminology, 107–109
- Weak interactive consistency, 36–44
 - abstractions, 37–40
 - algorithm, 40–42
 - certificates, 42–44
 - generally, 36–37
 - total order broadcast, 33–34
- Weighted voting framework, dependable systems, 92–94
- Wireless sensor networks, 115–142, 275–284
 - faulty network, 132–135
 - energy balance, 135
 - permanent faults, 132
 - transient faults, 132–135
- low-power control channel, 125–131
 - described, 126–128
 - performance, 128–131
 - variant in packet transfer latency, 131
- model definition, 117–119
- motivation and background, 276–279
- multichannel network, 135–139
 - fault-free network, 137–139
 - system model, 135–136
- overview, 115–117, 275–276
- permutation routing in single-hop networks, 121–125
 - fault-free network protocol, 121–123
 - fault-tolerant permutation routing, 123–124
 - remarks on, 124–125
 - system model, 121
- service centric model, 279–283
 - compared, 282–283
 - described, 280–282
 - generally, 279–280
 - system model, 119–120
- WIRM system architecture, 352–355
 - challenges, 354–355
 - generally, 352–354
- Xception approach, dependability
 - evaluation, fault injection, 329–331
- XOR gate, combinational circuit equivalence, 9